

EUROPEAN LANGUAGE GRID

D3.1

Requirements and Design Guidelines

Authors:	Julija Melnika (TILDE), Andis Lagzdinš (TILDE), Uldis Siliņš (TILDE), Raivis Skadins (TILDE), Andrejs Vasiljevs (TILDE)
Dissemination Level:	Public
Date:	30-06-2019

About this document

Project	ELG – European Language Grid
Grant agreement no.	825627 – Horizon 2020, ICT 2018-2020 – Innovation Action
Coordinator	Dr. Georg Rehm (DFKI)
Start date, duration	01-01-2019, 36 months
Deliverable number	D3.1
Deliverable title	Requirements and Design Guidelines
Type	Report
Number of pages	103
Status and version	Final – Version 1.0
Dissemination level	Public
Date of delivery	Contractual: 30-06-2019 – Actual: 30-06-2019
WP number and title	WP3: Grid Platform – Interactive Interface and Information System
Task numbers and titles	Task 3.1: Requirements collection and framework selection Task 3.2: Design, development and deployment of ELG front-end
Authors	Julija Melnika (TILDE), Andis Lagzdīns (TILDE), Uldis Siliņš (TILDE), Raivis Skadins (TILDE), Andrejs Vasiljevs (TILDE)
Reviewers	Gerhard Backfried (SAIL), Jose Manuel Gomez Perez (EXPSYS)
Consortium	Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany Institute for Language and Speech Processing (ILSP), Greece University of Sheffield (USFD), United Kingdom Charles University (CUNI), Czech Republic Evaluations and Language Resources Distribution Agency (ELDA), France Tilde SIA (TILDE), Latvia Sail Labs Technology GmbH (SAIL), Austria Expert System Iberia SL (EXPSYS), Spain University of Edinburgh (UEDIN), United Kingdom
EC project officers	Philippe Gelin, Alexandru Ceausu
For copies of reports and other ELG-related information, please contact:	DFKI GmbH European Language Grid (ELG) Alt-Moabit 91c D-10559 Berlin Germany Dr. Georg Rehm, DFKI GmbH georg.rehm@dfki.de Phone: +49 (0)30 23895-1833 Fax: +49 (0)30 23895-1810 http://european-language-grid.eu © 2019 ELG Consortium

Table of Contents

Table of Contents	3
List of Figures	5
Abbreviations	8
Abstract	9
1 Introduction	9
2 User Requirements Elicitation	9
2.1 User Survey	10
2.1.1 Targeted Groups	10
2.1.2 Survey Design	10
2.1.3 Survey Results	11
2.2 Interviews with User Representatives	31
2.2.1 Interview Goals	31
2.2.2 Interview Methodology	31
2.2.3 Primary Target Groups/User Personas	32
2.2.4 Findings	33
3 MVP User Journeys	34
3.1 Representative User Journeys	34
3.1.1 'Browsing for the Solution' User Journey	35
3.1.2 'Searching for the Solution' User Journey	36
3.2 Initial site structure	38
3.3 Specific pages	38
3.3.1 Homepage	38
3.3.2 Category Silo Pages	39
3.3.3 Subcategory Silo Pages	39
3.3.4 Search Results Page	40
3.3.5 Landing Pages	40
3.3.6 Content Page	40
3.3.7 Information Architecture	41
3.4 Content guidelines	41
3.4.1 Search Engine Optimisation	41
3.4.2 Future Content Development	42
3.4.3 Keeping in Touch through Newsletters	42
3.5 Sketches	42
3.5.1 Mobile Sketches	42
3.5.2 Desktop Sketches	47
3.6 Accessibility	52
4 Design Guidelines	53
4.1 Material Design	53
4.2 Design System and Interface Components	54
5 Front-end Architecture	75
5.1 Architecture Overview	75

5.2	Front-end Technology Stack	76
5.2.1	Choosing the Right Framework	76
5.2.2	Single Page Application	77
5.2.3	Theming Graphical Components	79
5.2.4	Tools for Development	79
5.2.5	Tools for Building and Packing	80
5.3	Content Management System	81
5.3.1	CMS Selection	81
5.3.2	Drupal Architecture	82
5.3.3	Selected Drupal Components	82
5.3.4	Menu Generation and Serving	84
5.3.5	Content Serving	85
5.4	API Requirements	85
5.5	Deployment of the Front-end	86
Annexes		87
Annex I. User Survey		87
Annex II. Interview Notes		97

List of Figures

Figure 1. Answers to question 1 ‘What is your organisation type’	11
Figure 2. Answers to question 2 ‘What is your primary role at the organisation?’	12
Figure 3. Answers to question 3 ‘In which of the following activities is your company/organisation involved?’	12
Figure 4. Answers to question 4 ‘Have you ever provided your language data (e.g. corpora, lexica, terminologies, etc.) or software/tools to any online platform (GitHub, META-SHARE, CLARIN, ELRA catalogue, etc.)’	13
Figure 5. Answers to question 5 ‘Have you ever submitted language data (datasets, corpora, etc.) through specific repositories/registries? Which ones?’	13
Figure 6. Answers to question 6 ‘Have you ever submitted software tools/services through specific repositories/registries? Which ones?’	14
Figure 7. Answers to question 7 ‘What types of content (data or software tools) have you submitted?’	14
Figure 8. Answers to question 8 ‘What was your motivation when submitting content?’	15
Figure 9. Answers to question 9 ‘How is your content (data/software tools or services) documented when shared? Do you use any specific metadata standards or schemas? If yes, which ones?’	15
Figure 10. Answers to question 10 ‘Have you experienced any difficulties in sharing your content?’	16
Figure 11. Answers to question 11 ‘What do you like about the platform in which you shared your content?’	16
Figure 12. Answers to question 12 ‘What parameters are the most important for you to measure the success of sharing your content? (select up to 3 most important)’	17
Figure 13. Answers to question 13 ‘How important for you are the following factors when submitting your content?’	17
Figure 14. Answers to question 14 ‘Are you interested in the option to provide your software as a container?’	18
Figure 15. Answers to question 15 ‘If you would like to provide a containerized software, what kind of support would you need? (multiple selection possible)’	18
Figure 16. Answers to question 16 ‘Would you be ready to pay for support that gives advice on how to best integrate your software in ELG?’	19
Figure 17. Answers to question 17 ‘Would you be ready to pay for support that can create a container out of your code?’	19
Figure 18. Answer no question 18 ‘If you plan to provide a containerized tool/software, would you prefer to:’	20
Figure 19. Answers to question 19 ‘Would you want to provide a paid content (to sell) through ELG?’	20
Figure 20. Answers to question 20 ‘How would you prefer to handle the payment owed to you (select all that apply)’	21
Figure 21. Answers to question 21 ‘Would you be ready to pay a fee to a platform that sells your software?’	21
Figure 22. Answers to question 22 ‘Have you ever searched language data or software/tools on such platforms as META-SHARE, ELRC-SHARE, CLARIN, ELRA etc.?’	22
Figure 23. Answers to question no 23 ‘What types of content (data/software) have you searched for? (select all that apply)’	22
Figure 24. Answers to question 24 ‘Where have you searched for language data (datasets, texts, etc.)? Please select all that apply.’	23
Figure 25. Answers to question 25 ‘Where have you searched for language processing software tools/services? Please select all that apply.’	23

Figure 26. Answers to question 26 ‘Please choose the most important criteria you use when searching for language data (e.g. datasets, corpora, lexica, terminologies, etc.). Select up to 5.’	24
Figure 27. Answers to question 27 ‘Please choose the most important criteria you use when searching for language processing software tools/services (select up to 5).’	24
Figure 28. Answers to question 29 ‘What would you prefer for using the software service/tool? (select all that apply)’	25
Figure 29. Answers to question 30 ‘How important is it for you to have the older versions of the same content (data/software) available on the ELG?’	25
Figure 30. Answers to question 31 ‘Have you experienced any difficulties in searching for language content on existing platforms (META-SHARE, ELRA, CLARIN, GitHub, etc.)? Please select all that apply.’	26
Figure 31. Answers to question 32 ‘Would you be willing to contribute the following types of language content?’	27
Figure 32. Answers to question 33 ‘Would you be interested to use the following types of language content?’	27
Figure 33. Answers to question 34 ‘How important would it be for you to have one centralized digital meeting spot for LT in Europe?’	28
Figure 34. Answers to question 35 ‘Are you interested in paid content (data/software)?’	28
Figure 35. Answers to question 36 ‘What type of service would be the most useful for you (select all that apply)?’	29
Figure 36. Answers to question 37 ‘How useful would it be for you to have more information about trainings, events and job advertisements?’	29
Figure 37. Answers to question 38 ‘Would you agree to create a full profile for yourself and/or your organisation and populate it with information?’	30
Figure 38. Answers to question 40 ‘In 2020, ELG will announce two open calls for demonstrator projects, where partial funding for development of innovative applications/technologies will be offered. Would you be interested in (select all that apply):’	30
Figure 39. Agenda for User Journey	35
Figure 40. Browsing for a Solution User Journey	36
Figure 41 Searching for a Solution User Journey	37
Figure 42. Mobile Newsletter	42
Figure 43. Mobile Homepage	43
Figure 44. Mobile Search results page	43
Figure 47. Mobile Silo page	44
Figure 48. Mobile Silo secondary page	44
Figure 47. Mobile Filter page	45
Figure 48. Mobile Menu page	45
Figure 51. Mobile Item page	46
Figure 52. Desktop Newsletter	47
Figure 53. Desktop Homepage	48
Figure 54. Desktop Search page	49
Figure 55. Desktop Silo page	50
Figure 56. Desktop Silo secondary page	51
Figure 57. Desktop Item page	52
Figure 58. Colour Scheme	55

Figure 59. System Icons	56
Figure 60. Theme UI	57
Figure 61. Typography Scale	58
Figure 62. Backdrop	59
Figure 63. Banners	60
Figure 64. Buttons	61
Figure 65. Cards	62
Figure 66. Chips	63
Figure 67. Dialogs	64
Figure 68. FAB	65
Figure 69. Lists	66
Figure 70. Menus	67
Figure 71. Navigation Drawers	68
Figure 72. Progress Indicators	69
Figure 73. Selection Controls	70
Figure 74. Sliders	71
Figure 75. Snackbar	70
Figure 76. Tabs	72
Figure 77. Text Fields	73
Figure 78. Text Fields (cont'd)	74
Figure 79. The Front-end architecture of European Language Grid	75
Figure 80. The core of the chosen technology stack	77
Figure 81. JWT creation and use in data exchange	79
Figure 82. Testing and bundling tools	80
Figure 83. Example of the Drupal menu structure, JSON-HAL format	84
Figure 84. Example of returned content page by CMS	85
Figure 85. Code delivery and deployment strategy of ELG front-end	86

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CLARIN	Common Language Resources and Technology Infrastructure
CMS	Content Management System
CSS	Cascading Style Sheets
EFNIL	European Federation of National Institutions for Language
ELG	European Language Grid
ELRA	European Language Resource Association
ELRC	European Language Resource Coordination action of CEF AT DSI
FAB	Floating Action Button
FIBEP	Federation Internationale des Bureaux d'Extraits de Presse (World's Media Intelligence Federation)
GATE	General Architecture for Text Engineering
GPL	General Public Licence
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICT	Information and Communication Technologies
JWT	JSON Web Token
LT	Language Technologies
META	Multilingual Europe Technology Alliance
MVP	Minimum Viable Product
POS	Part of Speech
REST	Representational State Transfer
RISIS	Research Infrastructure for Science and Innovation Studies
SEO	Search Engine Optimisation
SME	Small and Medium Size Enterprises
SPA	Single Page Application
SSR	Server-Side Rendering
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
WYSIWYG	What You See Is What You Get

Abstract

This document describes the requirements and design guidelines for the European Language Grid (ELG) platform. In Section 2 we present how user requirements were obtained. User requirements were elicited using two different approaches: through a user survey and user interviews. For each of the approaches, the report describes and analyses the findings. Based on the latter, in Section 3 we present two user journeys for a minimum viable product (MVP) version of the ELG. Following the user journeys, we describe the site structure, content guidelines and sketches for the mobile and desktop versions. Section 4 contains detailed design guidelines and a description of the material design framework. In the last section, Section 5, we have provided the front-end architecture overview, the chosen technology stack, API requirements and front-end deployment description. A specific subsection is dedicated to the Content Management System (CMS).

1 Introduction

This deliverable is a follow up to the previously submitted deliverables D2.1 and D8.1. To further analyse the user requirements outlined in D2.1 “User Requirements and Functional Specifications”, we have conducted a broader user survey and a series of user interviews.

The graphical design principles and interface elements presented in this document are based on the common ELG visual identity provided in D8.1 “ELG Project and Initiative Identity”. This report explicitly uses this identity to develop the design guidelines for the whole ELG front-end. These generalised guidelines, thus, can be used in all graphic user interface (GUI) elements including the web interface for desktop and mobile devices, the ELG newsletters and other digital communication channels.

It should be noted that initially the first GUI release was planned for March 2020. The Consortium decided to have an earlier release, the MVP version¹ to be presented at META-FORUM 2019 conference in Brussels on 8/9 October 2019. Thus, we have concentrated on the functionality necessary for the MVP. The sketches and mock-ups included in this document are specifically prepared for the MVP version. In the next phases of the project, these designs will be further elaborated and extended to other functional interfaces of the ELG platform. The MVP version will be used to collect user feedback to further improve the whole ELG platform including the front-end functionality, interface and design.

2 User Requirements Elicitation

In the requirement elicitation process, we have used the agile approach, which is especially suitable for long-term new product development projects where it is neither possible nor feasible to define and describe the whole system in the starting phase of the project. In D2.1 “User Requirements and Functional Specification”, we have defined the initial user groups and identified and described the main user scenarios for each group. In

¹ Described in detail in Deliverable 1.2 Base Infrastructure (first release).

the next step, we need to enhance our understanding of user needs and expectations, their existing pain points² and unobtained gains. Two key instruments are used for this purpose – a user survey and the user interviews.

The user survey has been used to gain initial responses about:

- user experience
- feedback on existing practices
- factors influencing significant decisions, e.g., the type of support needed for creating and/or using containerised tools. To simplify responding to the survey, open ended questions were avoided; however most of the questions did have an option for the respondents to add free-form comments.

Users were interviewed to obtain additional insights into respondents' motivations, pains and gains, their emotional insights, and ideas. Interviews have followed a semi-structured approach. This approach establishes a core structure of questions, while allowing deviations from the existing question scenarios and following the natural flow of conversation.

2.1 User Survey

2.1.1 Targeted Groups

The survey was aimed at both the research and business communities involved with Language Technologies. The survey was constructed so that respondents could skip the content provision or content search question branches in case they had no relevant experience. However, the respondent list was devised with the goal that each respondent would go through at least one of the two abovementioned branches. The following respondent groups were invited to participate:

- META-NET
- Cracking the Language Barrier
- LT-Innovate
- ELRA
- EFNIL
- CLARIN
- ELRC
- GATE community
- ICT-29b projects
- RISIS research infrastructure
- SoBigData research infrastructure
- FIBEP (world medial intelligence federation)

The invitation to participate was shared through email campaigns and through the personal social network accounts of members of the consortium.

2.1.2 Survey Design

The survey design covered the following topics:

- Professional background of the participant

² Pain point is a marketing term for a specific problem that prospective clients/users are experiencing.

- Existing experience with submission of language data or language processing software tools/services
 - Submission of containerised software
 - Submission of paid content
- Existing experience with searching for language data or language processing software tools/service
- General questions
- Contact information (optional)

The survey consisted mainly of multiple-choice questions (one possible answer) or questions allowing several answers. Some of the questions used the Likert scale from 1 to 4 to estimate specific factors. Open-ended questions were generally avoided, though every question had a field for additional comments. The list of survey questions with answer options can be found in Annex I.

2.1.3 Survey Results

This section details the collected survey results and findings. The survey ran from the 27 May 2019 until 24 June 2019, a total of 158 respondents participated. We also used branching in the survey, e.g., if in Q4 the user answered that they had no experience of submitting resources, then the following group of questions dedicated to resource submission (Q5-Q21) was skipped. The partial responses from 50 participants who did not finish the survey are not included in the analysis, which is based on completed responses from 108 respondents.

The majority of respondents (67%) said that they represent the academic/research segment (Figure 1). The industry segment consisting of Industry/Large, Industry/SME and Start-ups was represented by 24% of the answers. Based on the comment provided, we have also included the person who chose 'Other' in the industry segment group. In the industry segment, 54% of the respondents represent SMEs, 23% are start-ups and 19% represent large industry players. In our following analysis, we review the differences between the answers given by these academic/research and industry groups, in cases where the answers differ significantly.

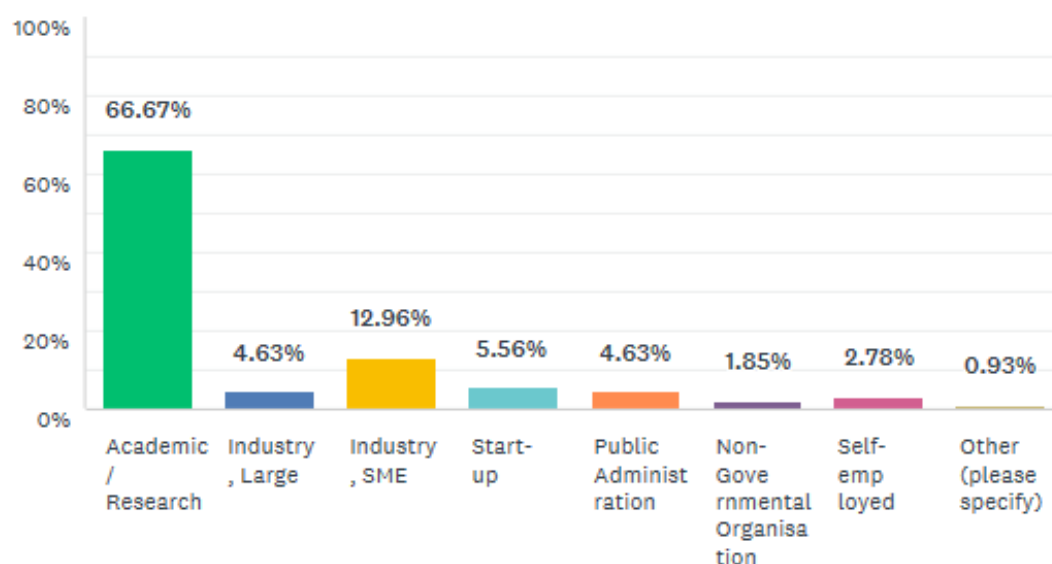


Figure 1. Answers to question 1 'What is your organisation type'

Out of the 108 responses received, the majority (52%) said that their primary role in their organisation is Researcher (Figure 2). Developer, Research/Academic Manager and Business Owner are among the most mentioned answers.

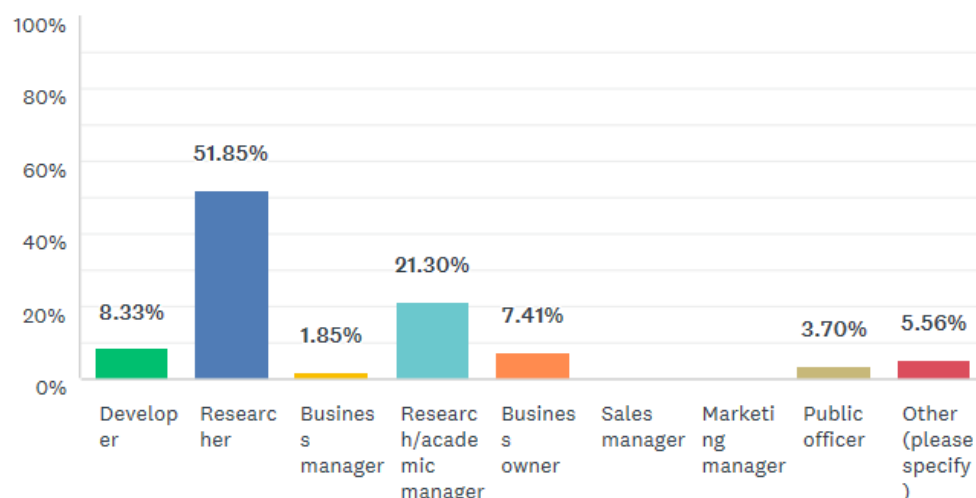


Figure 2. Answers to question 2 'What is your primary role at the organisation?'

When we asked about the activities in which the company/organisation is involved (Figure 3), the most popular answers were Language technology research (for the academic/research group) and Language technology development (for the industry group). Apart from that, Academic teaching in natural language processing has been popular with academic/research, and Data Analytics has been popular with industry. Both groups also often mentioned Language technology services.

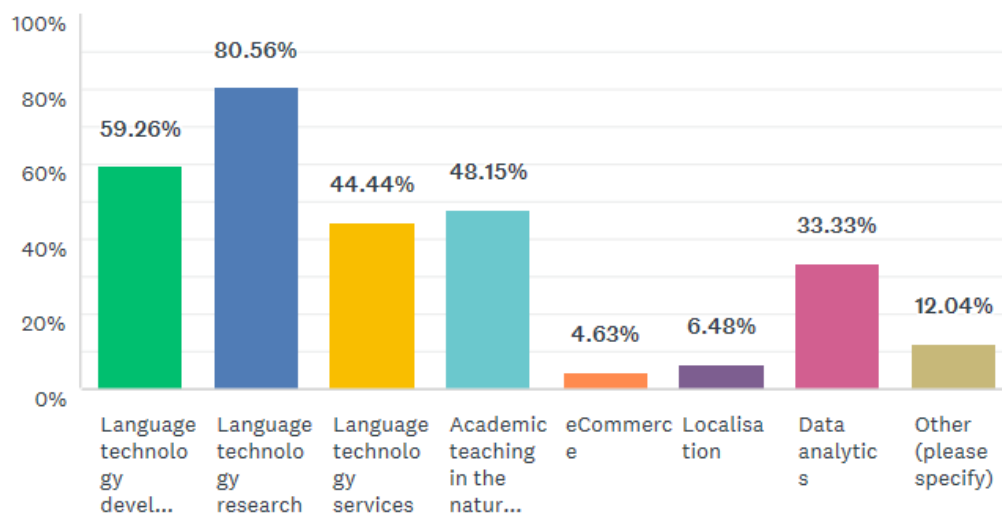


Figure 3. Answers to question 3 'In which of the following activities is your company/organisation involved?'

The majority of respondents (61%) said that they have had the experience of providing their language data or software/tools to an online platform (Figure 4). It is interesting that while the academic/research group has answered predominantly positive (78% answered yes), the industry group has mostly not provided language data or software/tools before (54% answered no).

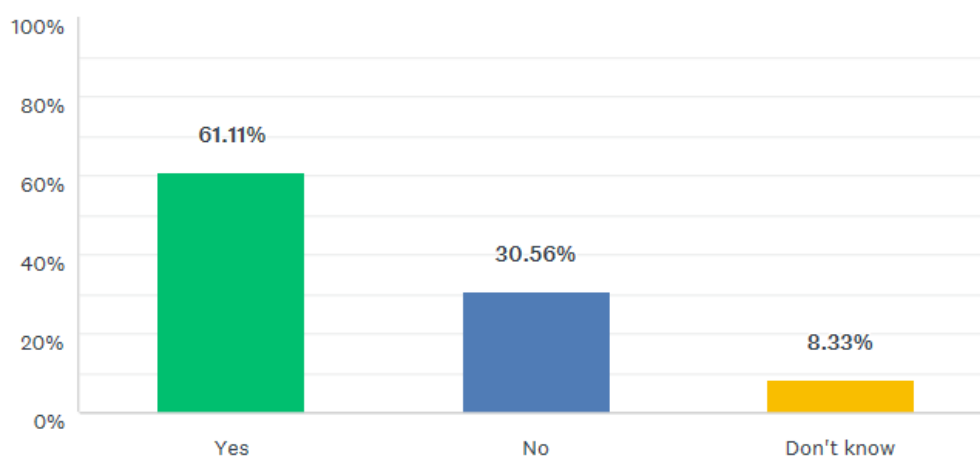


Figure 4. Answers to question 4 'Have you ever provided your language data (e.g. corpora, lexica, terminologies, etc.) or software/tools to any online platform (GitHub, META-SHARE, CLARIN, ELRA catalogue, etc.)'

The next two questions asked about specific repositories where data (Figure 5) or tools/resources (Figure 6) were provided. In both questions, the answers with the top percentage were Institutional repository, Institutional/personal website (first place for the academic/research group regarding language data) and Software repository such as GitHub, gitlab, etc. (first place for the academic/research group regarding software/tools and first place for the industry group in both questions). CLARIN repository was the third most popular option.

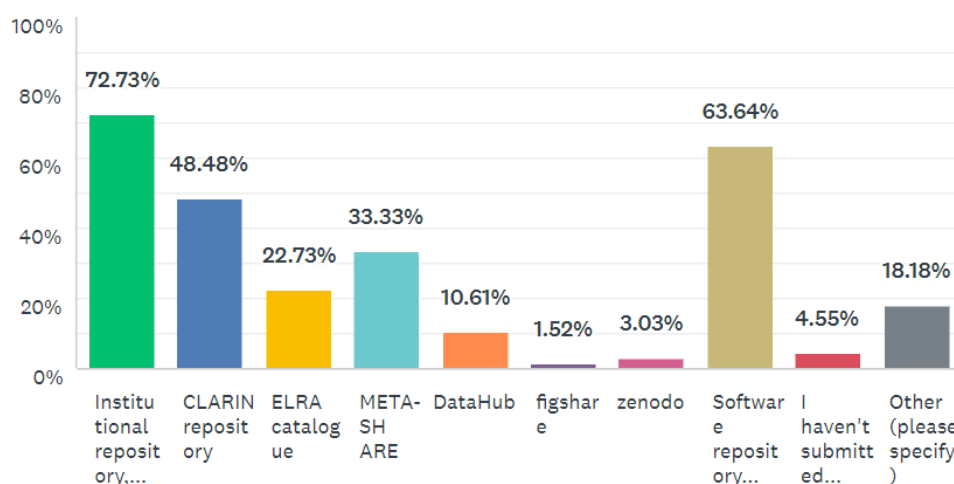


Figure 5. Answers to question 5 'Have you ever submitted language data (datasets, corpora, etc.) through specific repositories/registries? Which ones?'

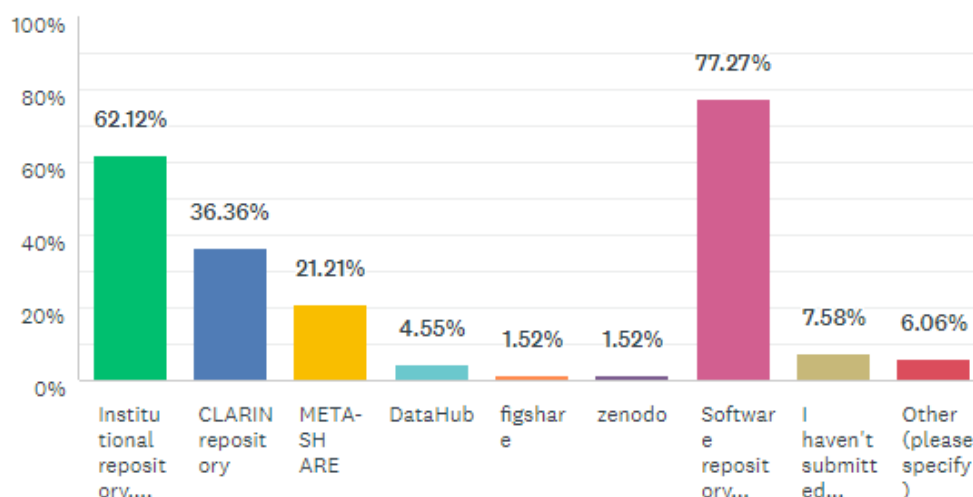


Figure 6. Answers to question 6 'Have you ever submitted software tools/services through specific repositories/registries? Which ones?'

In the next question, we asked about the specific type of content that has been submitted by respondents (Figure 7). The industry group has confidently chosen the Tools/Services option (88% against 44% for next biggest answer). The academic/research group has also put Tools/Services in first place (89%) but without such prominent majority. Around 70% of the academic/research group has also submitted textual datasets/collections of text data/corpora and Annotated datasets/ Collections/ Corpora.

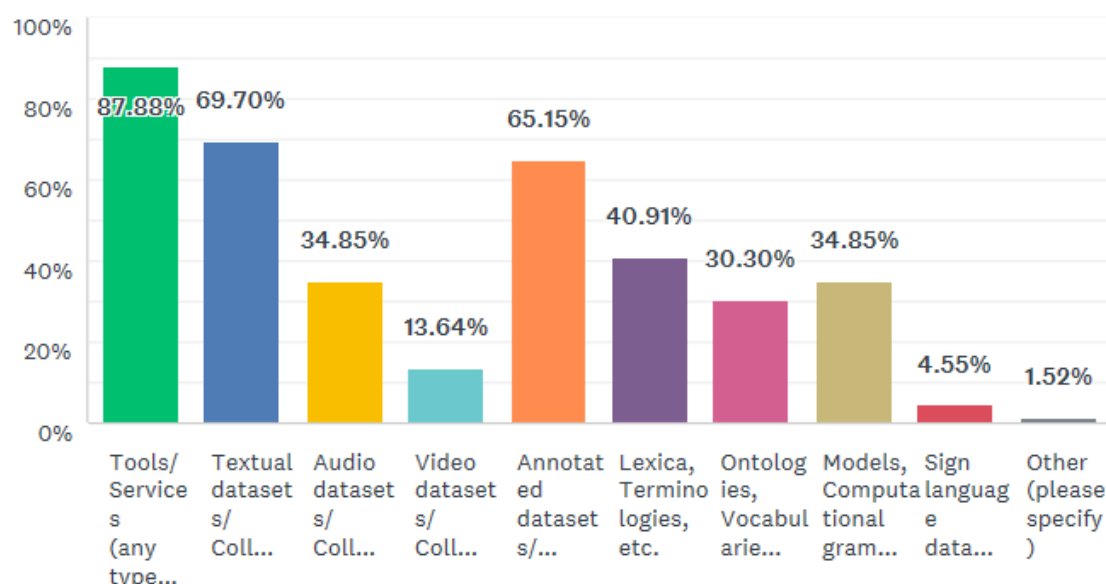


Figure 7. Answers to question 7 'What types of content (data or software tools) have you submitted?'

One of the most interesting questions for us was the motivation of content providers (Figure 8). Both the industry and academic/research groups agreed that promotion of the content is the most important motivation. However, their next chosen options were strongly different. The academic/research group is also motivated by goodwill (serving the community), needs to ensure the reproducibility of research results and has project requirements as motivation. Meanwhile, the industry group cares very much about promotion of the organisation.

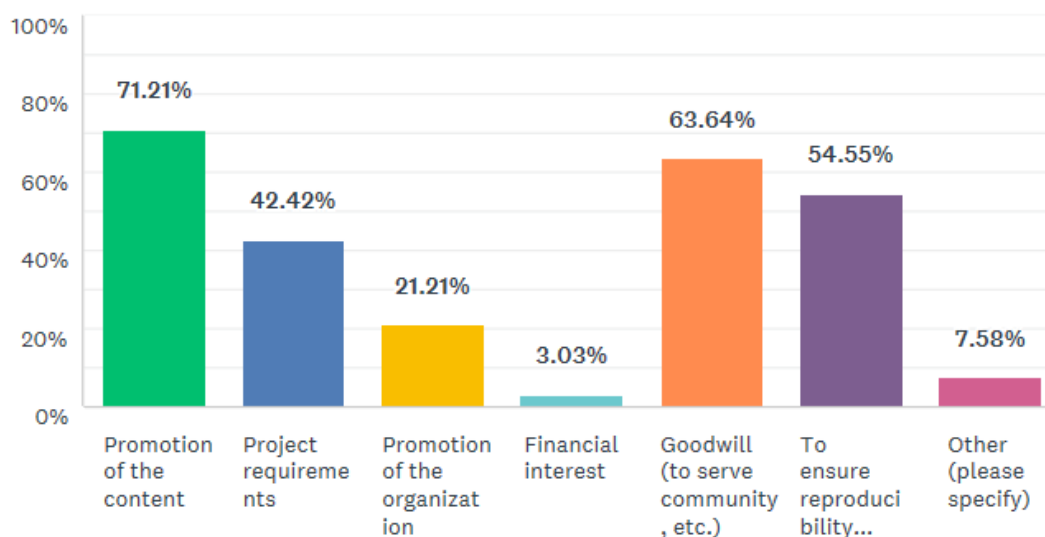


Figure 8. Answers to question 8 ‘What was your motivation when submitting content?’

In the next question, we asked about the documentation standard for data/software tools/services (Figure 9). The predominant answer is that Metadata standards/schema were not necessary, which puts quite an educational challenge on the ELG. META-SHARE and CMDI profile were mentioned as the second and third options.

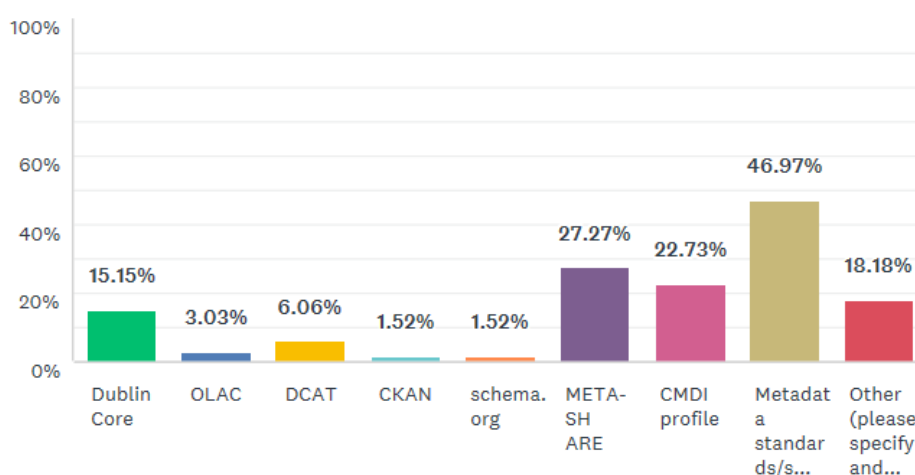


Figure 9. Answers to question 9 ‘How is your content (data/software tools or services) documented when shared? Do you use any specific metadata standards or schemas? If yes, which ones?’

While answering if they have experienced any difficulties in sharing content (Figure 10), the academic/research group mentioned Copyright/IPR issues as the biggest problem. The industry group has chosen the choice of license to assign as the biggest problem. As the third biggest problem, both groups have identified that the process was too time-consuming. For a clear view of all the answer options, please see Annex I.

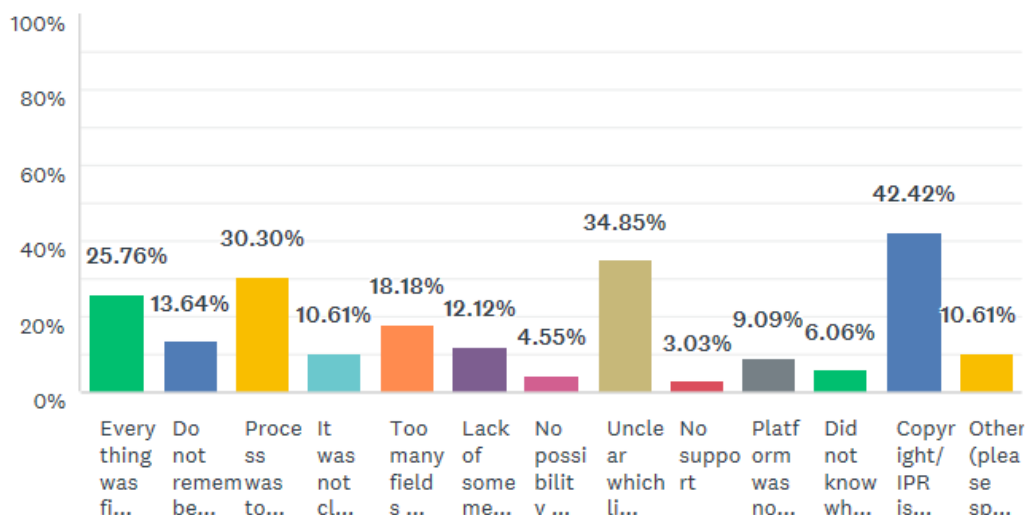


Figure 10. Answers to question 10 'Have you experienced any difficulties in sharing your content?'

The next question asked the opposite – the positive side of the platforms on which content was shared. The majority has identified that the experience was positive in general (53%) and that the platform is well-known/well-recognised (52%). In fact, the second answer is extremely important for the industry group.

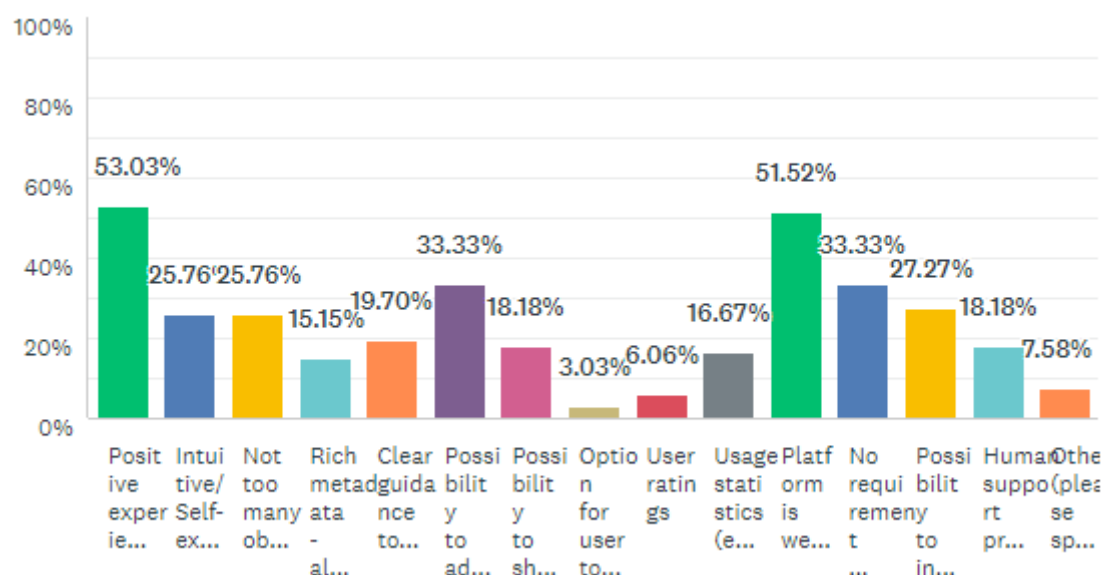


Figure 11. Answers to question 11 'What do you like about the platform in which you shared your content?'

Continuing the theme of motivation, we have enquired about the parameters of success (Figure 12). Number of downloads and references in publication were chosen as the most important parameters (the first for the industry group, the second for the academic/research group).

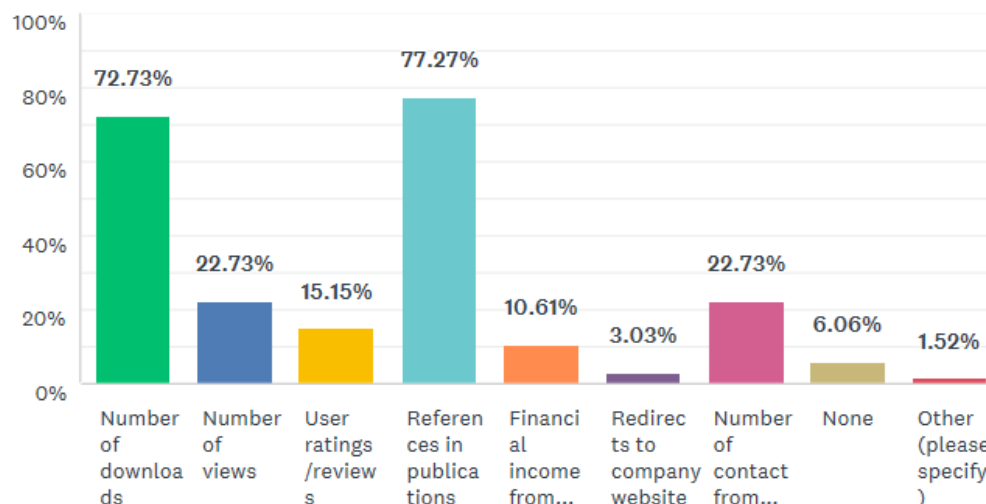


Figure 12. Answers to question 12 'What parameters are the most important for you to measure the success of sharing your content? (select up to 3 most important)'

In the next question, we wanted to evaluate the importance of nine specific factors for resource submission (Figure 13). It is interesting to know that the simplicity of the submission process is extremely important for all

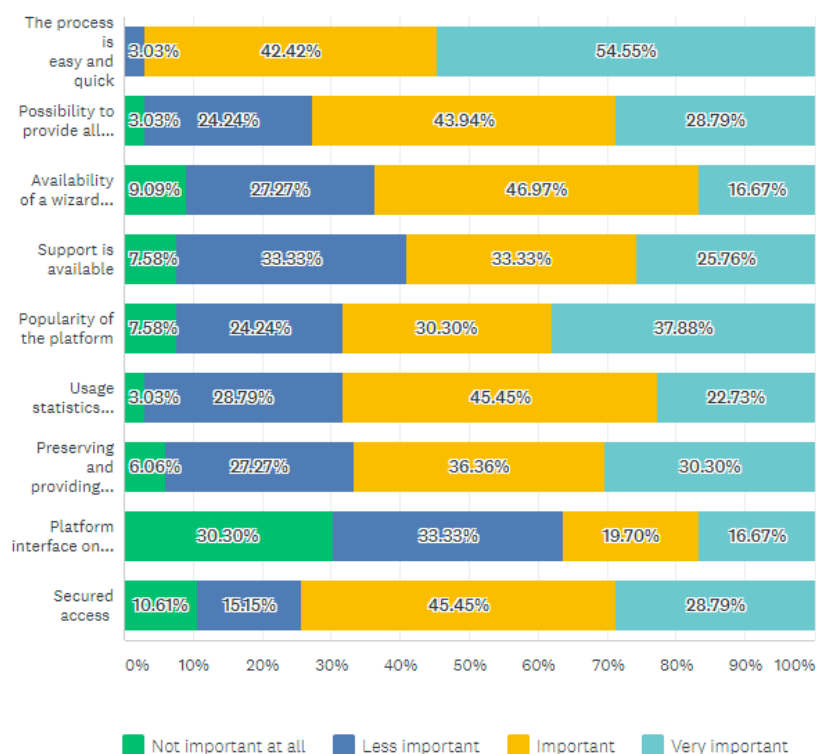


Figure 13. Answers to question 13 'How important for you are the following factors when submitting your content?'

users (none in the industry group marked it as less important or not important at all!). Secured access has also been marked as important for resource provision.

Both the academic/research and industry groups are interested in providing software as a container (Figure 15).

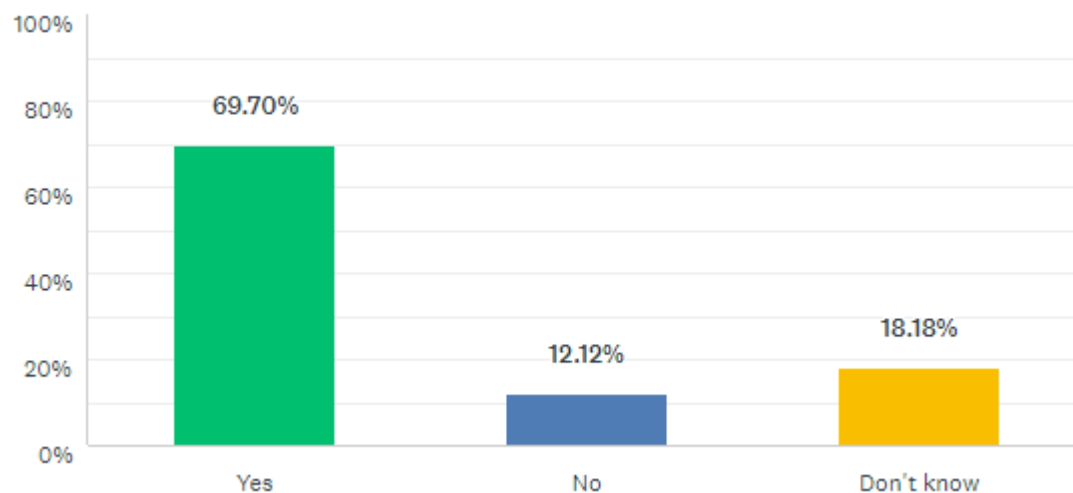


Figure 15. Answers to question 14 'Are you interested in the option to provide your software as a container?'

In the next group of questions, we wanted to know what kind of support would users prefer (Figure 14 to Figure 18). Written instructions (87%) and email support (48%) were chosen as the two main options that users would want.

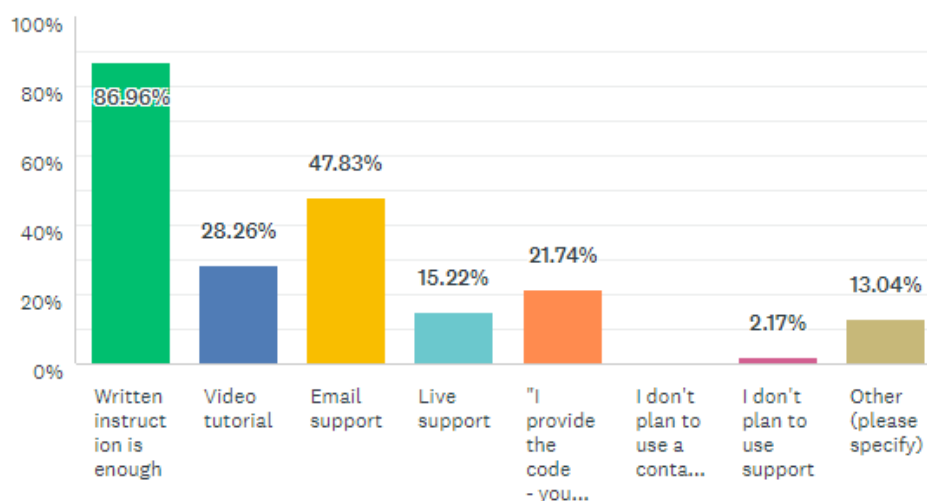


Figure 14. Answers to question 15 'If you would like to provide a containerized software, what kind of support would you need? (multiple selection possible)'

Accordingly, when we asked if users would be ready to pay for support that gives advice on how to best integrate software into ELG (Figure 16) and support that can create a container out of code (Figure 17), the answers are mostly no. Although the industry group in general is ready to pay for advice on software integration.

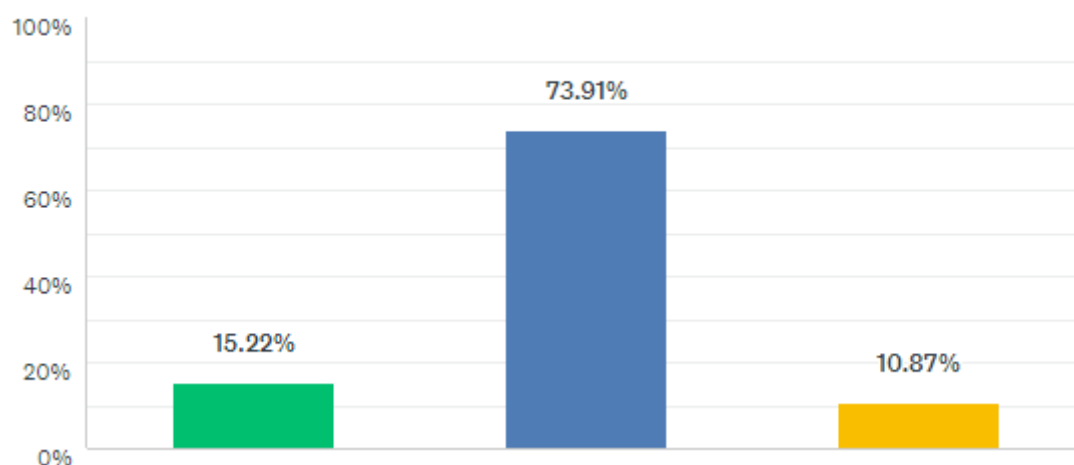


Figure 16. Answers to question 16 'Would you be ready to pay for support that gives advice on how to best integrate your software in ELG?'

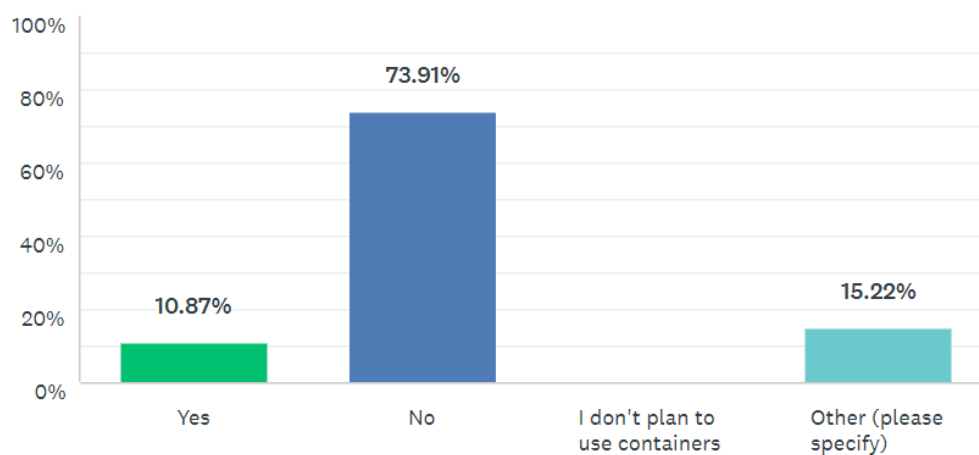


Figure 17. Answers to question 17 'Would you be ready to pay for support that can create a container out of your code?'

Next, we asked how the providers would like to provide the containerised tool/software (Figure 18). The top answer is a link to a remote repository where software is hosted (65%). The academic/research group also agrees to deploy software through the ELG platform.

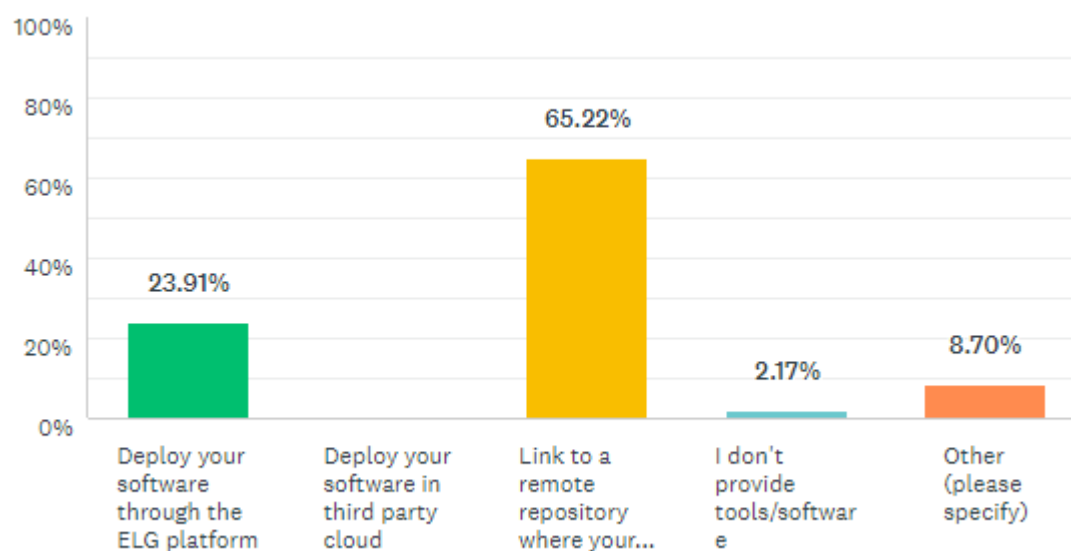


Figure 18. Answer no question 18 'If you plan to provide a containerized tool/software, would you prefer to:'

When asked if providers would want to sell their content through ELG (Figure 19), the academic/research group replied with a rather strict no, while the industry group has an equal number of respondents for and against.

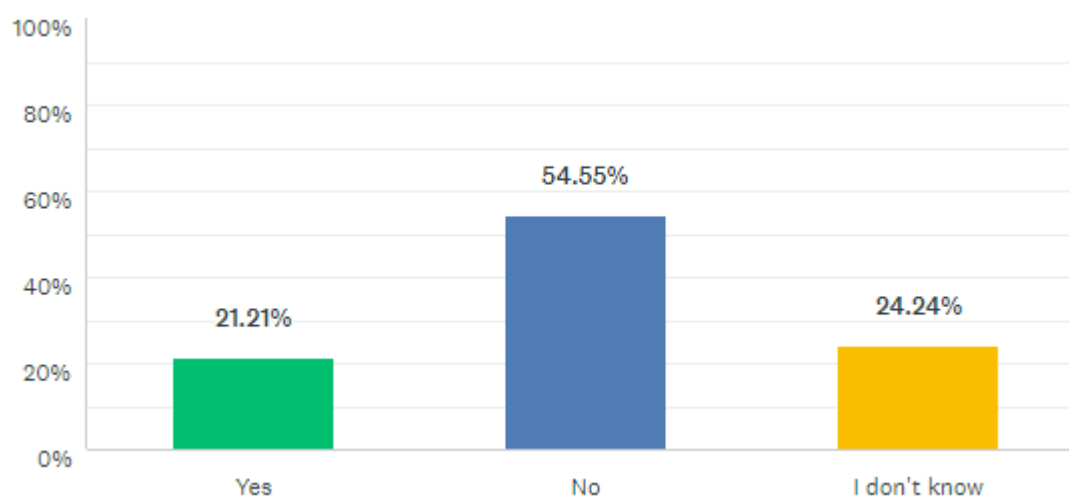


Figure 19. Answers to question 19 'Would you want to provide a paid content (to sell) through ELG?'

The next logical question was to find out how the providers would prefer to handle payments (Figure 20). And again, we had an equal number of responders for two options: by themselves or through ELG as an intermediary.

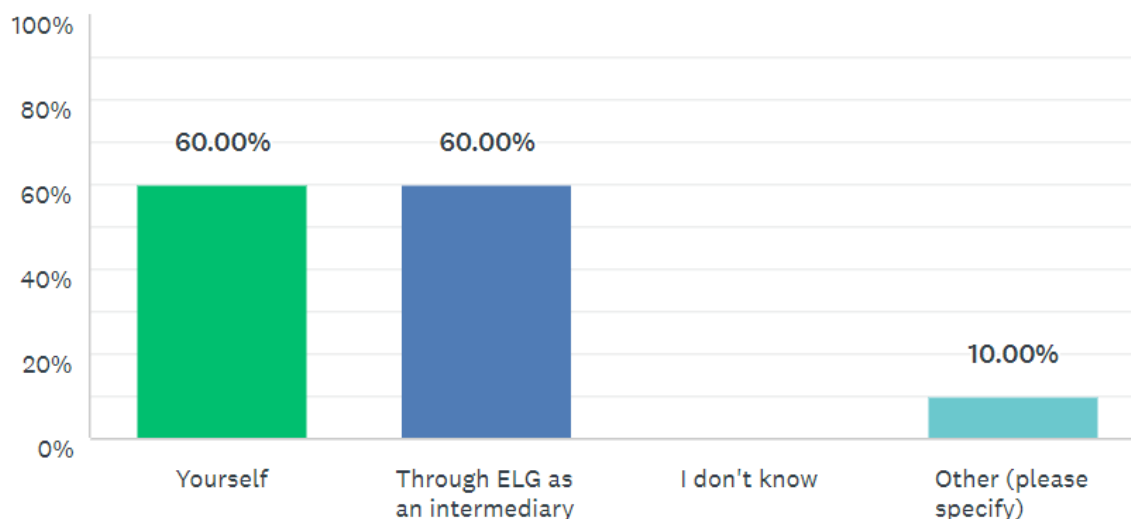


Figure 20. Answers to question 20 'How would you prefer to handle the payment owed to you (select all that apply)'

Almost 36% said that they would be ready to pay a 10-20% fee to a platform that sells their software, 29% said that they would be ready to pay only up to 10% and 14% would not want to pay at all to a platform that sells their software (Figure 21). The academic/research group prefers the up to 10% option, while the industry group is ready for a 10-20% fee.

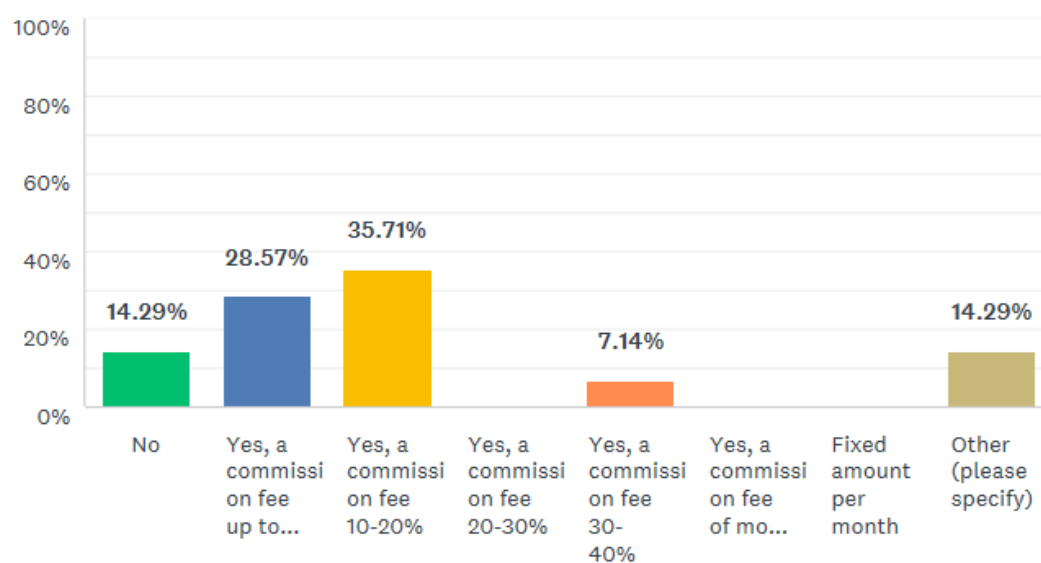


Figure 21. Answers to question 21 'Would you be ready to pay a fee to a platform that sells your software?'

The next big group of questions was dedicated to language data or software/tool search.

72% of respondents said that they have searched language data or software on platforms such as META-SHARE, ELRC-SHARE, CLARIN, ELRA, etc., while 25% said that they have no such experience (Figure 22).

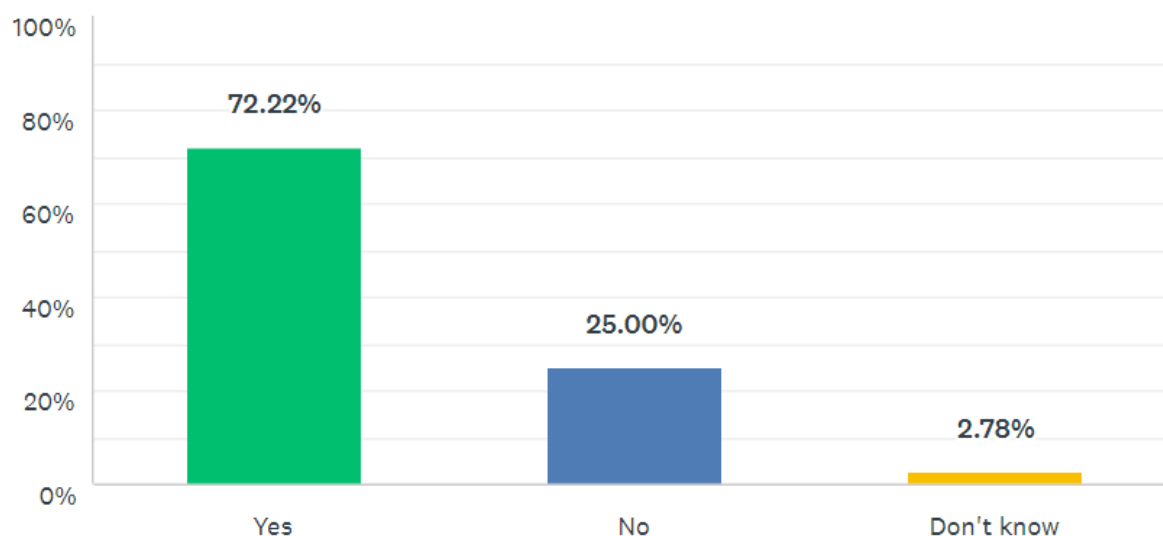


Figure 22. Answers to question 22 'Have you ever searched language data or software/tools on such platforms as META-SHARE, ELRC-SHARE, CLARIN, ELRA etc.?'

What is extremely interesting is that the biggest part of our respondents has been looking for Textual datasets/ Collections of text data/ Corpora (87%) and Annotated datasets/ Collections/ Corpora (73%). Tools/services take only the third place for the industry group and fourth place for the academic/research group (Figure 23).

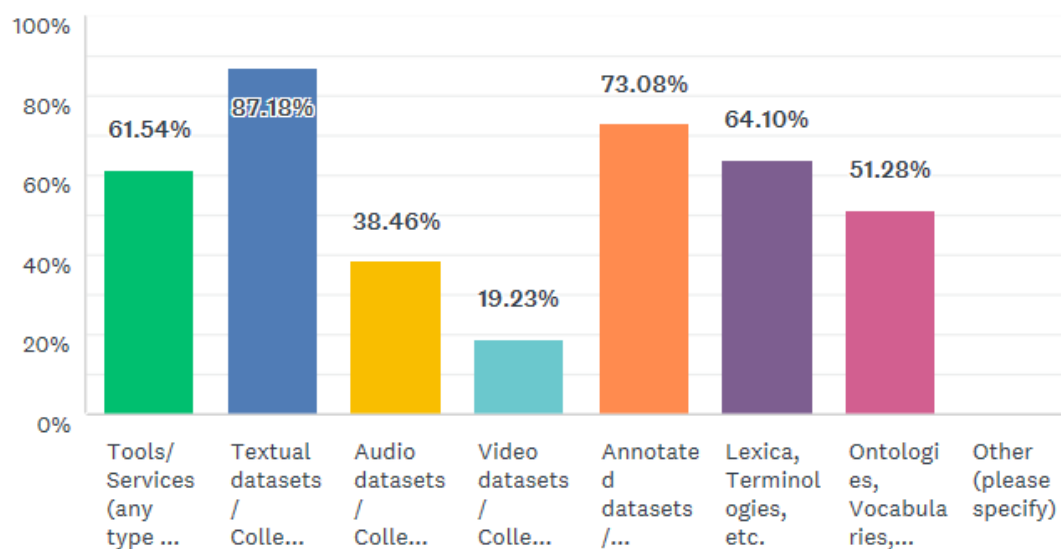


Figure 23. Answers to question no 23 'What types of content (data/software) have you searched for? (select all that apply)'

Language data is mostly searched for on such platforms as CLARIN (64%) or using a general search engine (63%). ELRA catalogue, Software repository (GitHub/GitLab etc.) and LDC (Linguistic Data Consortium) all share third place with 56%. When analysed separately, the industry group has interesting specifics. For them, the main search places for language data are LDC (Linguistic Data Consortium) and ELRA catalogue (Figure 24).

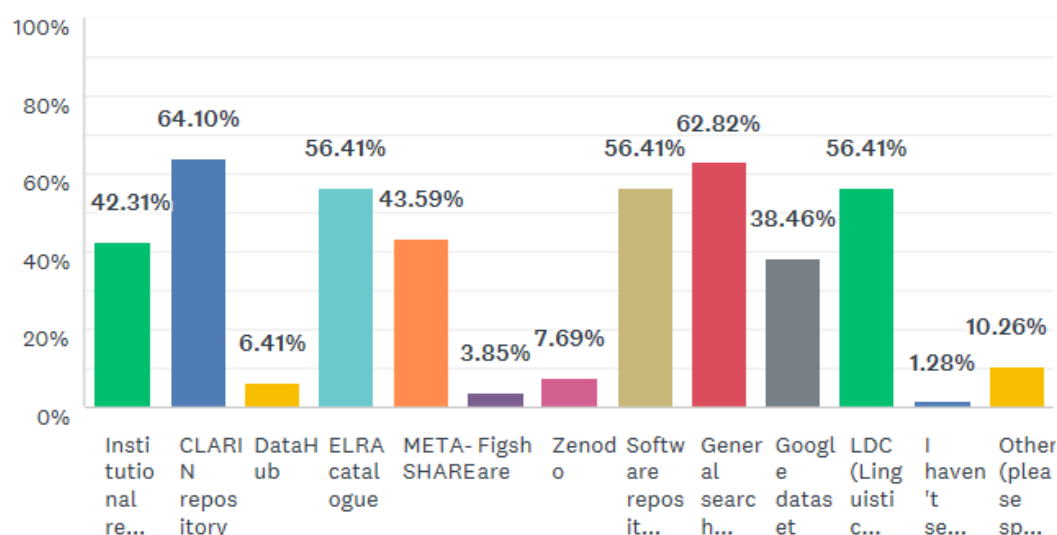


Figure 24. Answers to question 24 'Where have you searched for language data (datasets, texts, etc.)? Please select all that apply.'

Language processing software/tools are mainly searched for through Software repositories (GitHub/GitLab/docker hub, etc.) and a general search engine. Results are the same for both groups. This shows how important SEO (search engine optimisation) is for a platform such as ELG.

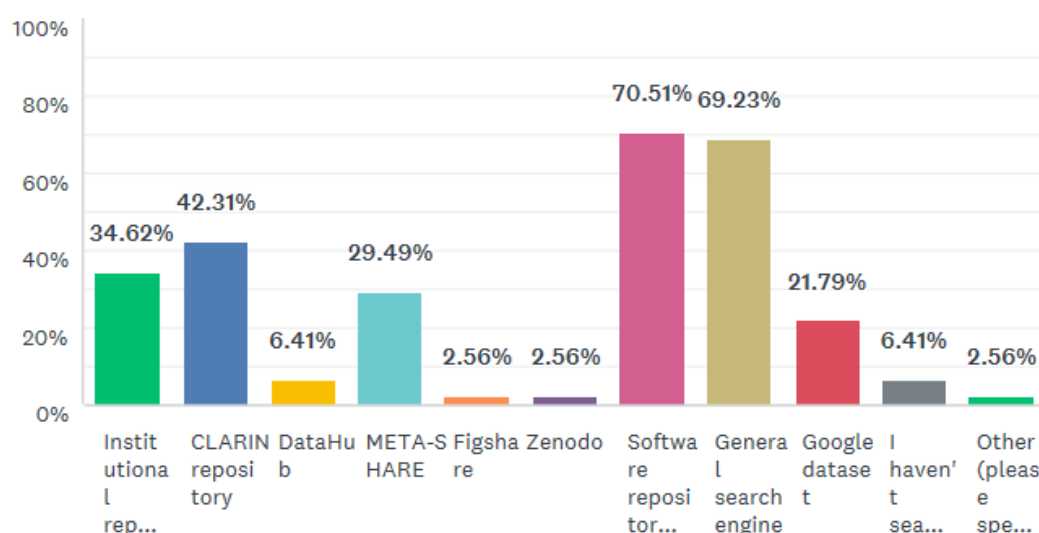


Figure 25. Answers to question 25 'Where have you searched for language processing software tools/services? Please select all that apply.'

In the next two questions, we asked respondents to state the most important criteria when they are searching for language data or language processing software/tools.

When searching for language data (Figure 26), the academic/research group has chosen Language of the contents (81%) and License/access conditions (71%) as the most important. For the industry group, the most important criteria are Domain/topic/type of data (e.g., finance, politics, biology, tweets, etc.) (94%) and Language of the contents (88%).

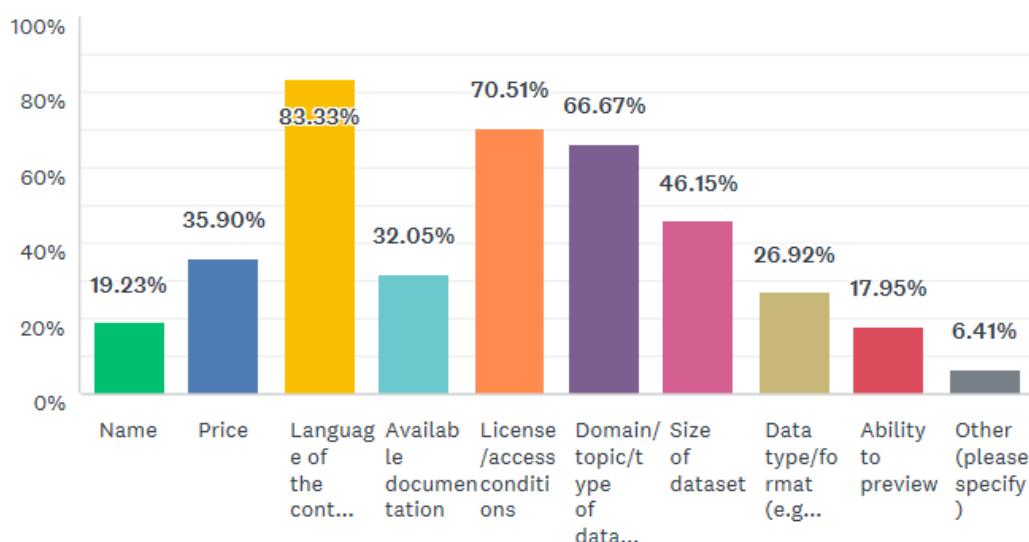


Figure 26. Answers to question 26 'Please choose the most important criteria you use when searching for language data (e.g. datasets, corpora, lexica, terminologies, etc.). Select up to 5.'

For language processing software tools/services, the most important criteria are:

- For the academic/research group: Availability of open source code (67%) and Language coverage (60%)
- For the industry group: License/access conditions (75%) and Language coverage (68%)

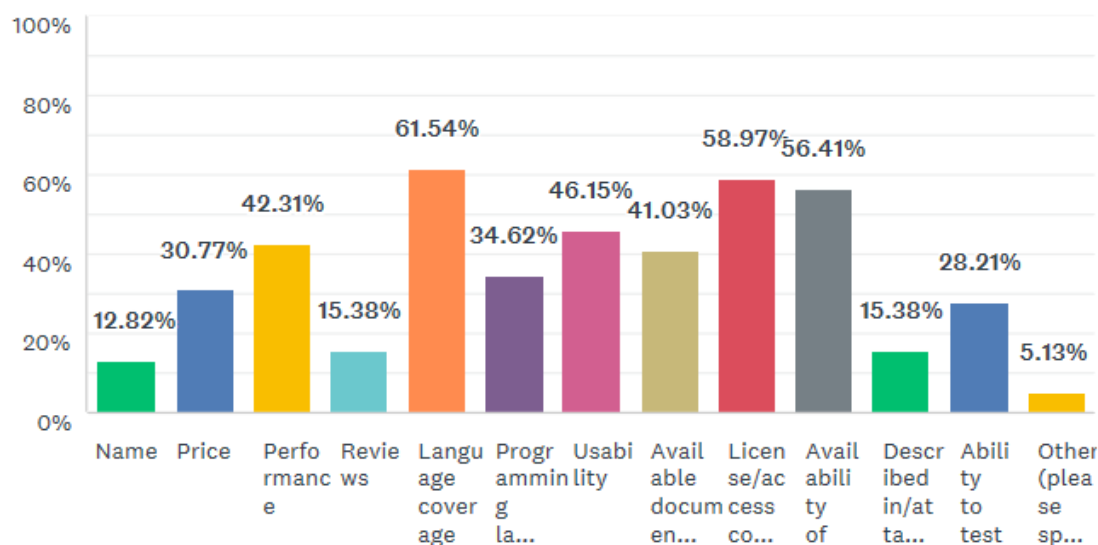


Figure 27. Answers to question 27 'Please choose the most important criteria you use when searching for language processing software tools/services (select up to 5).'

Question 28 is skipped in our analysis, is it provided purely for referential value.

When it comes to using software services and tools, 72% would prefer to download executables or code and 63% prefer to download containerised tools/services (Figure 29). For the industry group, both options are equally attractive.

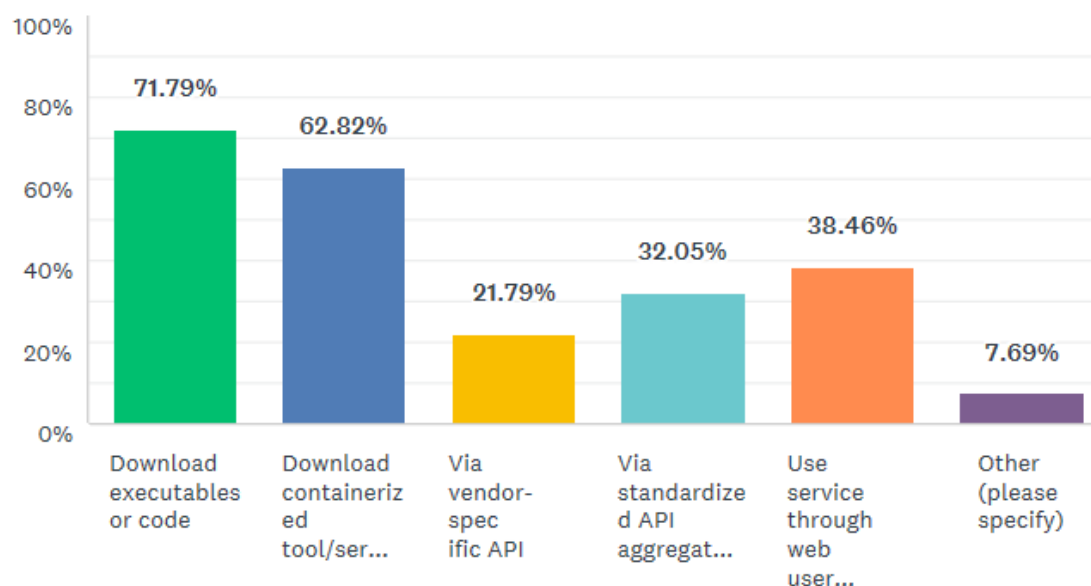


Figure 29. Answers to question 29 'What would you prefer for using the software service/tool? (select all that apply)'

In our next question, we wanted to know if respondents need to have older versions of data/software (Figure 28). The answers differed: the academic/research group considers this to be rather essential (43% important), while the industry group does not really care (56% of less important).

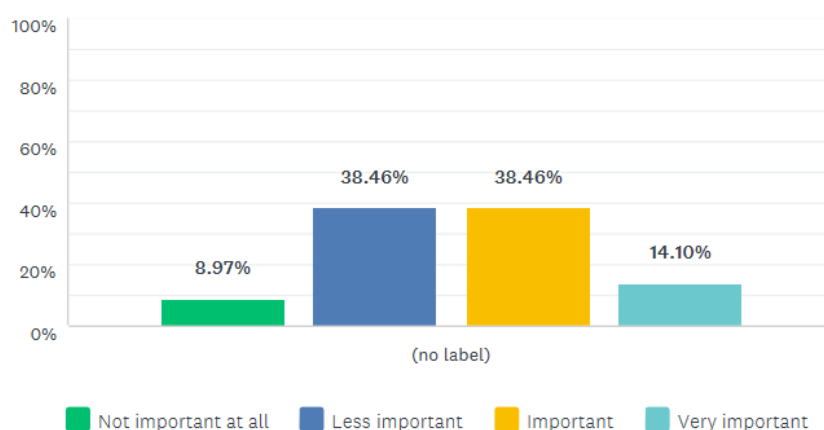


Figure 28. Answers to question 30 'How important is it for you to have the older versions of the same content (data/software) available on the ELG?'

The same is true with providers; we wanted to hear about factors disturbing search according to users' experience (Figure 30).

- The academic/research group mentioned problems with finding necessary data/software (49%), too many irrelevant results, insufficient description and/or metadata information, and lack of samples/demo/trial (all 36%).
- The industry group chose unclear licence and usage conditions (44%), problems with finding necessary data/software (37%) and lack of samples/demo/trial (31%)

It is interesting to see that while some of the problems are general for everybody, everyone struggles with finding the specific result that they want. However, there are also pretty specific problems for each group.

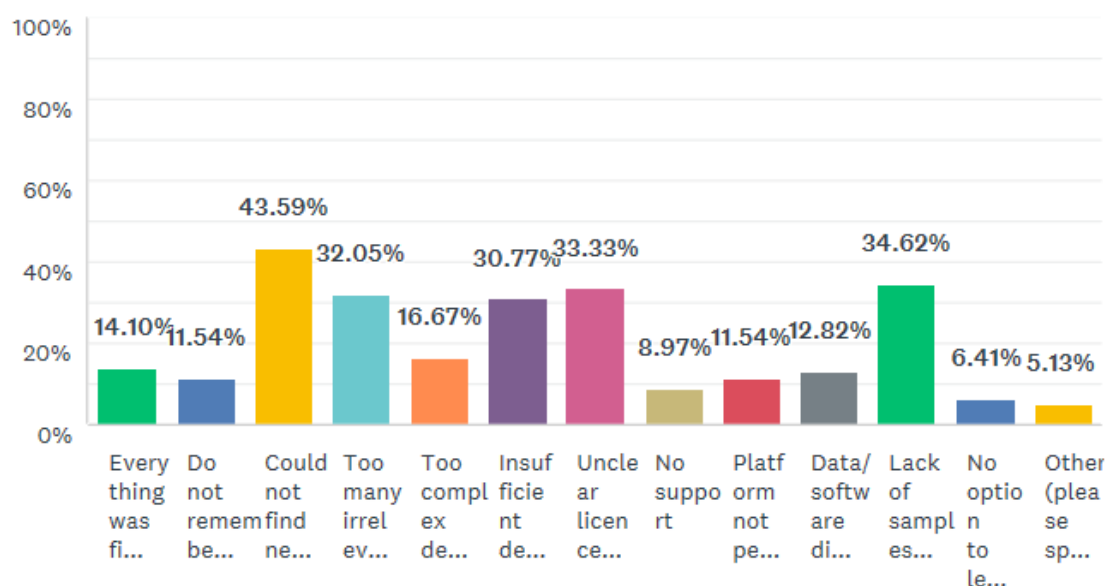


Figure 30. Answers to question 31 'Have you experienced any difficulties in searching for language content on existing platforms (META-SHARE, ELRA, CLARIN, GitHub, etc.)? Please select all that apply.'

With that, the branch for resource seekers is finished, and the next group of questions was shown to everybody.

First, we wanted to know about respondents' willingness to contribute the various types of language-technology-related content (Figure 31). The academic/research group is rather generous with 73% of positive answers for corpora and tools or services. The industry group is ready to offer tools or services (53%), with corpora taking second place (34%).

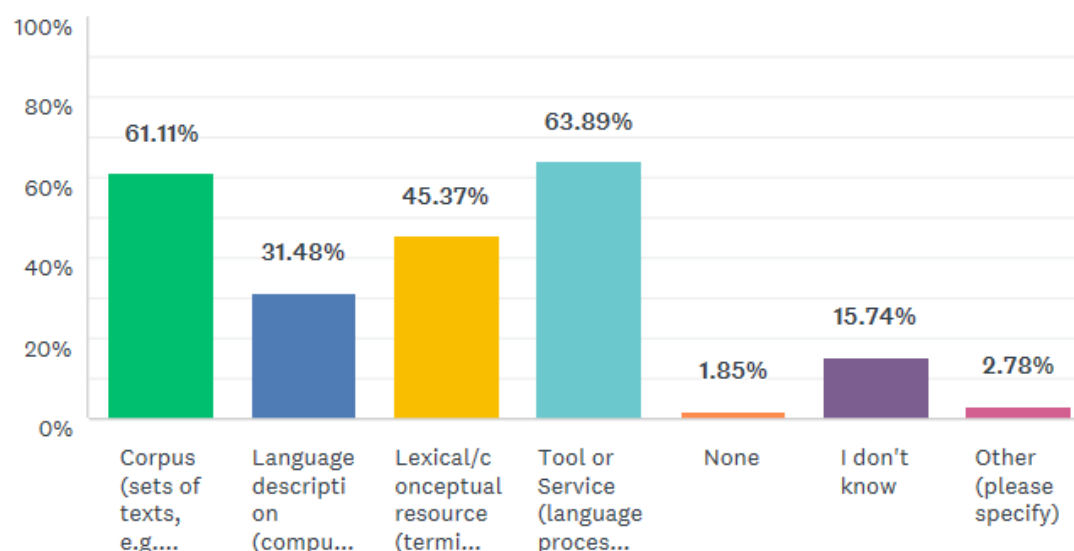


Figure 31. Answers to question 32 'Would you be willing to contribute the following types of language content?'

When asked about interest in usage of content (Figure 32), both groups confirmed their interest for the same two types of content, although the academic/research group is a bit more interested in tools or services (97%), while the industry group needs data sets and corpora more (80%).

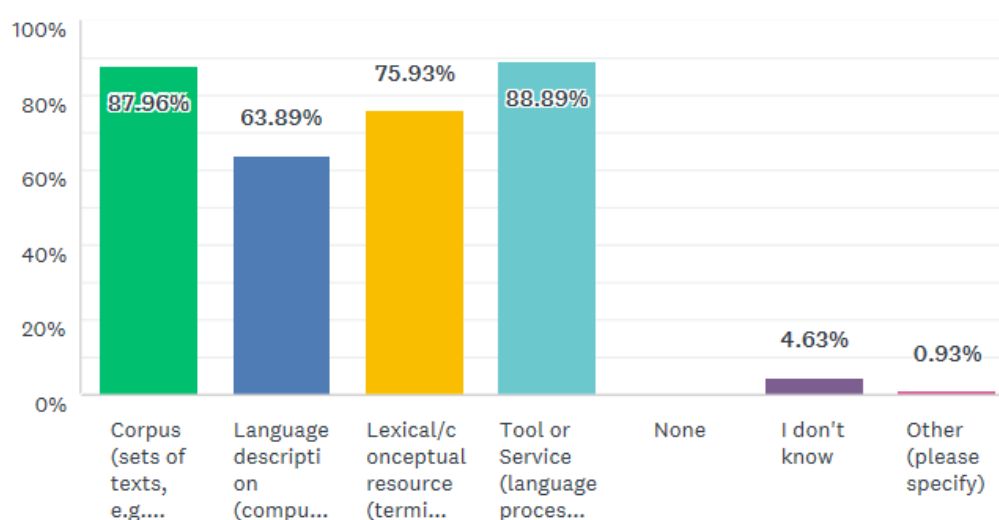


Figure 32. Answers to question 33 'Would you be interested to use the following types of language content?'

Both groups agreed that it is important (47%) to have one centralised digital meeting spot for LT in Europe (Figure 33).

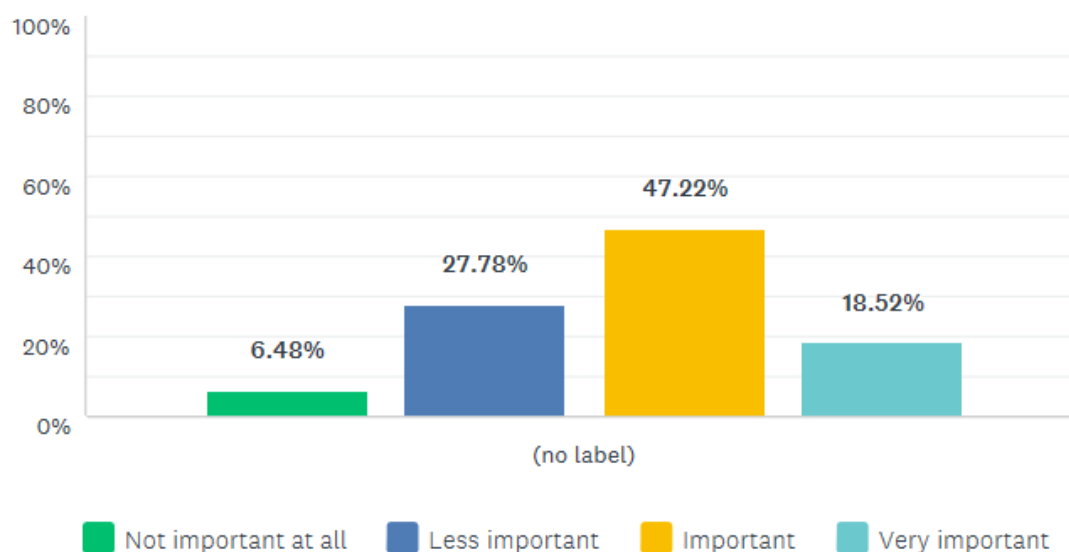


Figure 33. Answers to question 34 'How important would it be for you to have one centralized digital meeting spot for LT in Europe?'

When asked about interest in paid content (Figure 35Figure 34), opinions again were divided. The academic/research group is mostly searching only for free content (58%), and only 23% are ready to pay. On the other hand, the industry group is quite ready to pay for good content (65%).

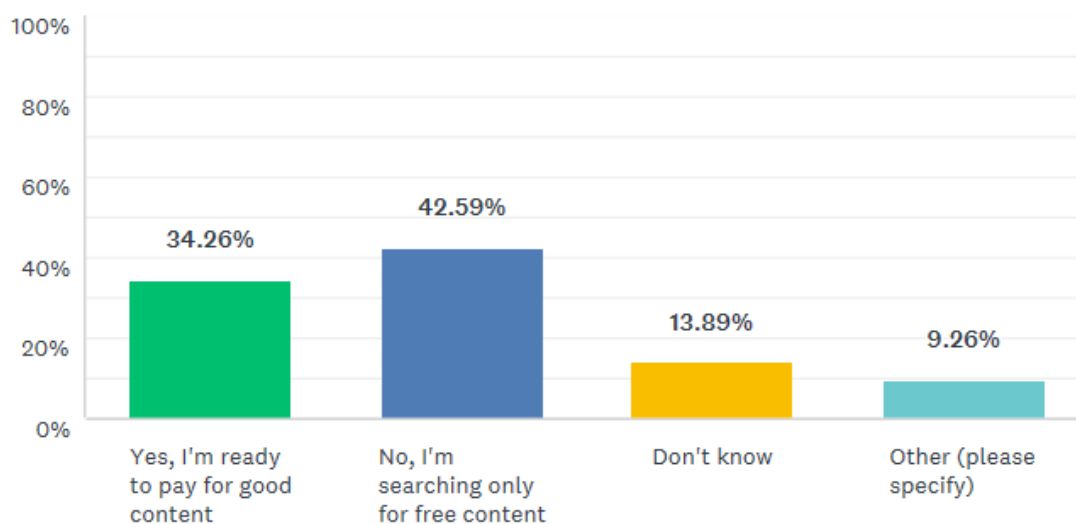


Figure 34. Answers to question 35 'Are you interested in paid content (data/software)?'

We asked respondents if they would prefer the synchronous or asynchronous request approach, but there is no clear answer here as both groups unanimously decided that both approaches are necessary (Figure 35).

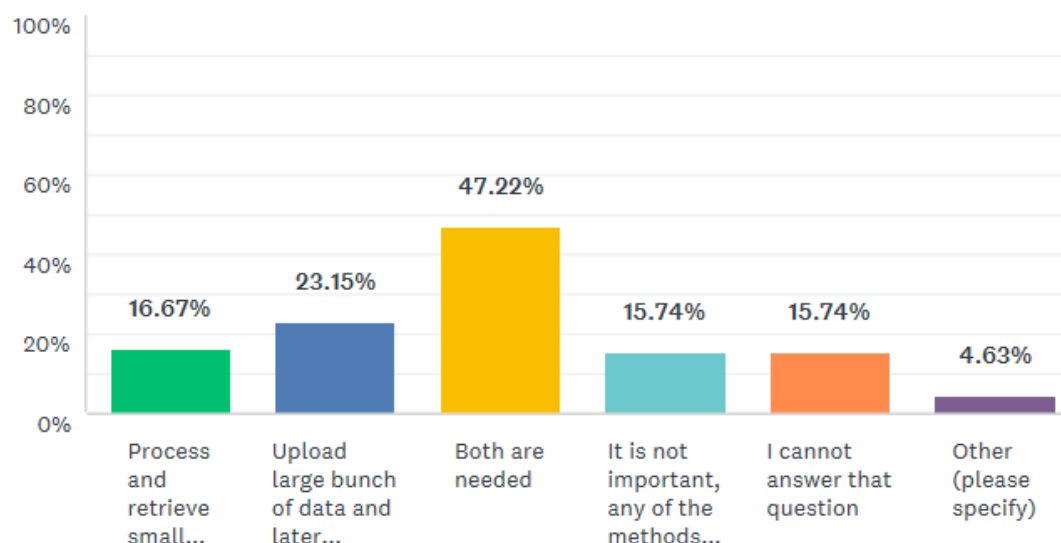


Figure 35. Answers to question 36 'What type of service would be the most useful for you (select all that apply)?'

Both groups stated that training and event information is rather important to them (more than 50% rated it as important or very important). However, job advertisement information was mostly rated as not very important (Figure 36).

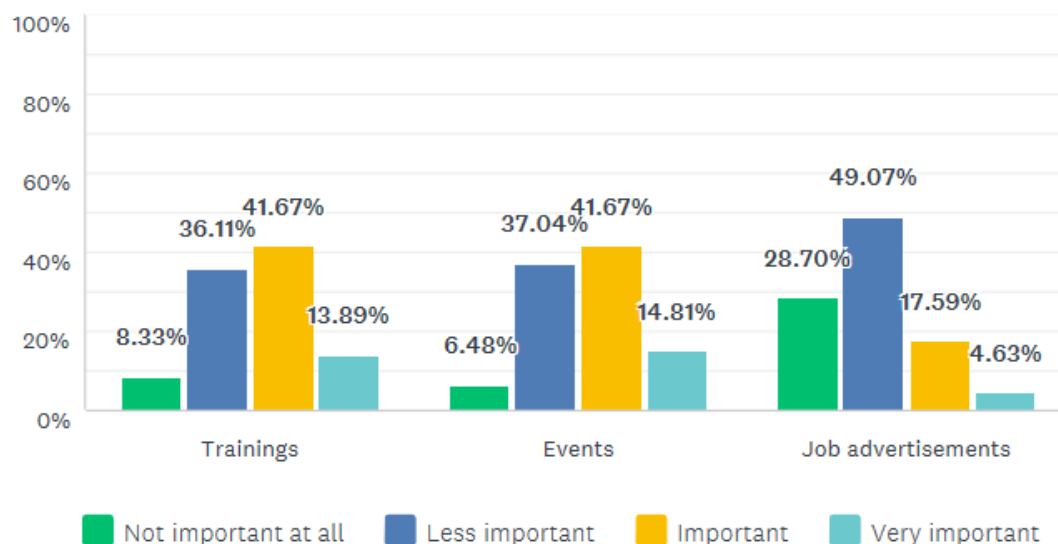


Figure 36. Answers to question 37 'How useful would it be for you to have more information about trainings, events and job advertisements?'

We asked users if they would want to create their profile on ELG with full information (Figure 37). The academic/research group was rather positive about that (59% yes), while the industry group is rather hesitant, with approximately 45% for both 'yes' and 'I don't know' options. It looks like the industry group is waiting to see how appropriate the platform will be in relation to its needs.

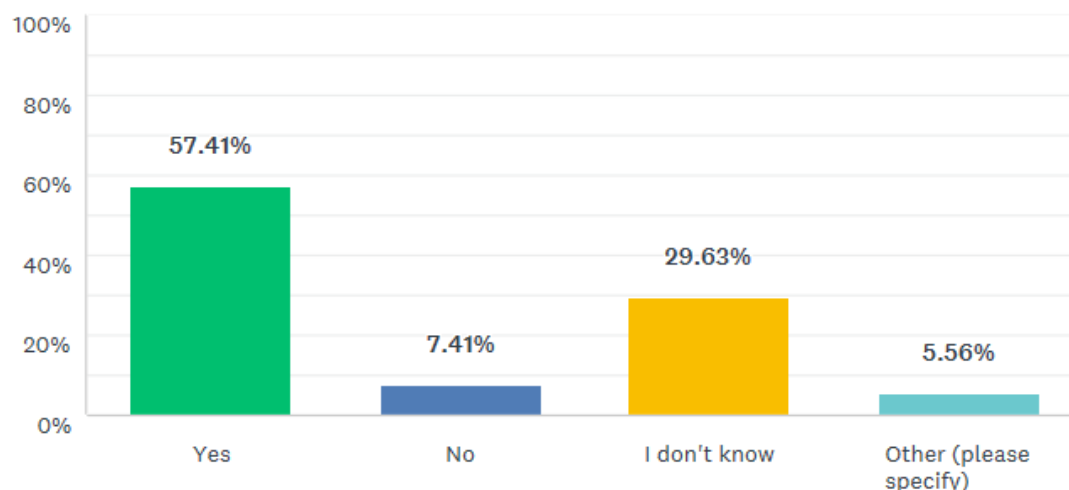


Figure 37. Answers to question 38 'Would you agree to create a full profile for yourself and/or your organisation and populate it with information?'

Question 39 was skipped due to its descriptive nature.

In the next question, we provided information about future open calls for demonstrator projects and asked if respondents would be interested in that. 63% would be interested to get more information, and 54% of the academic/research group is already ready to take part in the open calls and submit a project. Within the industry group, only 38% confirmed this option.

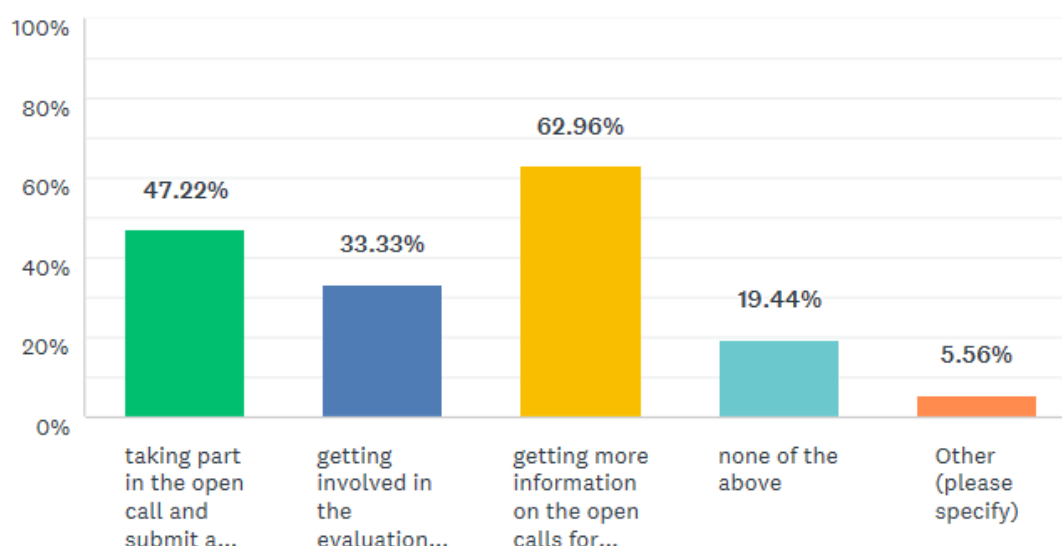


Figure 38. Answers to question 40 'In 2020, ELG will announce two open calls for demonstrator projects, where partial funding for development of innovative applications/technologies will be offered. Would you be interested in (select all that apply):'

2.2 Interviews with User Representatives

User interviews are a valuable method for the elicitation of user requirements that have not been identified by other means. At the moment, we have brainstormed a few target groups/user personas, who have different contexts, motivation and goals. In order to explore these primary users further, we performed six interviews. We were looking for qualitative rather than quantitative, statistically backed insights. However, some of the insights we have gathered will later be validated on broader user groups, possibly with additional quantitative evaluation. It is worth to state that this is an ongoing process that will also be employed during the next phases of the project as we explore further user stories. The first batch of target groups/user personas will give us a broad context that can be explored further.

2.2.1 Interview Goals

The primary goal of these interviews was to uncover:

- Context
- Needs
- Motivation / Expected outcome

To rephrase this: in what situations (context) does the user want to do something (need) so that he could achieve his goal(s) (motivation/expected outcome). We have employed the jobs-to-be-done³ methodology, which views products and solutions in terms of jobs a customer is trying to get done, i.e., it explores why a customer needs the product. By employing this methodology, we wanted to go beyond only needs and expected outcomes but also explore further and see the context and motivation behind this. In this case, we are looking for a broader context that would later help us to create a more user-centric product by allowing our team to understand and empathise with the user even more.

Secondary goals:

- Gaining a broader user profile, the outcomes users produce, how they measure the success of their daily work, what is their background.
- Understanding and exploring existing pain points and happy points in the context we intend to cover with ELG.
- Exploring their past experiences with existing tools and products.
- What are the typical issues users have and during which stages of their user journeys.
- Gathering insights that we have not considered otherwise.
- Sharing interview insights with the team, allowing us to gain a better insight into users' minds.

These insights will later be used in building customer journey maps and experience mapping exercises as we delve further into building the ELG solution.

2.2.2 Interview Methodology

- Some interviews were carried out face-to-face, others as virtual online meetings. Interviews lasted for around one hour, some up to one hour and a half.
- All interviews were recorded, and notes were taken during conversations. The interview recordings were later reviewed to avoid missing important details.
- Users were informed that interviews will be anonymised and recorded.

³ <https://jtbd.info/>

- The core questions were phrased in the same way as to retain consistency; however, follow-up questions were different for each person.

The core interview consisted of three parts:

1. Exploring the interviewee's background and experience with LT, understanding what outcomes he/she produces and how he/she measures success.
2. Exploring past experiences. We did not ask users what they wanted, as this would force users to come up with features that only they needed. Rather we asked users to recall relevant past experiences, if they had any, and to mention pain points and happy moments. In cases where a user had no previous experience, we tried to explore how ELG might be useful by asking about broader contexts and problems.
3. Pitching the idea of ELG and asking for feedback.

2.2.3 Primary Target Groups/User Personas

This list is based on a previously developed list of user groups and includes our assumptions about which users might represent our primary audience. At the moment, we can only assume the information we know about these users.

- **LT developers/integrators who are working in the LT industry, looking for tools/services.** During their workflow, they might be searching for a specific language tool for a specific application or language. For example, someone who needs a good tokenizer for the Croatian language.
- **LT developers/integrators who are working in the LT industry, looking for language data/resources.** These users already have a good set of tools and require good quality data in the specific language and specific dialect to train or improve their AI algorithm. For example, a company working in the speech recognition field might need good quality transcribed audio files in a specific Arabic dialect.
- **LT developers/integrators who are working in the LT industry, ready to provide language tools/resources.** These users have developed several tools or resources and see them as their competitive advantage. They are not ready to share them for free (although there are exceptions) and expect to receive financial gains from selling them. One notable exception is when the users need to provide tools/resources due to external obligations, e.g., according to requirements of received financing.
- **Non-LT developers/integrators who are working in an industry and might find LT useful, but it is not their primary product.** These users might not be experts in LT tools/resources; however, they do understand the output that they produce. For example, in order to broaden the target audience, a game developer is asked to find a tool that would automatically localise their game in all major European languages, i.e., such a user might not be that interested in specific tools, as much as in a turnkey solution. Since they are not familiar with LT tools, they might also be looking for training or support, so they could implement these solutions.
- **LT researchers who are doing research in the field, looking for tools/resources and submitting tools/resources.** These users are researchers who are mainly focusing their work in the LT field. Use cases might vary, but mostly these users are looking for tools or language data that might assist their research. Such users might be more motivated to submit a tool or resource for the general public, as they may be interested in gaining recognition and carrying on with research.
- **Non-LT researchers who are doing research outside of LT, looking for tools/resources that might be useful in their research.** These researchers are not familiar with the field of LT and consequently are

not likely to submit LT tools/resources for the broader public. However, LT tools/resources might assist them in the research they are doing. For example, someone who is doing social media analysis as a part of their sociology research project might find sentiment analysis tools useful. Thus, combining this with a data scraper would allow this user to create a tool to back up his research or provide good findings. Such users may not be aware of all the possibilities that LT can provide because their knowledge of LT terminology is limited. Since they are not familiar with LT tools, they might also be looking for training or support for implementing these tools.

- **Other professionals that are connected with LT but are not developers.** These users may have some understanding of LT but are not interested in a specific tool that might interest developers. They are more interested in finding out which companies are active in specific areas, which technologies allow to perform certain tasks and what the current state of development is for these technologies. These users might even be frequent visitors of LT-related events, their network might include authoritative LT professionals, they might rely on outsourced solutions, and they might be looking for the best solution in the market. Such users might also be interested in gaining a deeper understanding of LT, i.e., different consultants and government officials.
- **Other professionals that are not connected with LT and are not developers.** Such users might include business owners and digital product owners. These users are not familiar with LT terminology, nor do they possess an advanced understanding of how LT tools work. They may have a limited awareness and want to explore the possibilities. These users will not be that interested in specific tools (a Croatian tokenizer for example), but they might be looking for advice and outsourced solutions. These users don't represent companies that are developing LT tools.
- **Resource owners, familiar with LT.** For example, owners of large datasets that do understand that such data might be valuable for the LT industry. These users might even represent a third party that is the primary data owner, where the primary data owner is not aware of such use cases. Such users might be competent enough to provide the necessary meta tags and understand the legal issues when submitting such tools or resources.
- **Resource owners, not familiar with LT.** For example, these could be data owners who are not aware or might have a limited awareness that the data they own might be valuable to the LT industry. These users lack an understanding of the legal issues or how to organise their data. While they might rely on a third party to submit their data, they might seek more information online.

2.2.4 Findings

Six interviews were conducted with users from various target groups:

- two interviews with LT developers/integrators who are working in the LT industry and are looking for language data/resources
- one interview with a resource owner, familiar with LT
- two interviews with other professionals who are connected with LT and are not developers
- one interview with a professional who is not connected with LT and who is not a developer

The output from these interviews has been used to detail user journeys and to create a list of potential issues/pain points to resolve in the future. Interviews with other professionals were of special interest, as this group is not covered by the user survey.

The main findings of the interviews are as follows:

- Even with the large variety of various platforms offering LT, users still prefer to search using general search engines such as google.com. Very often, this is the starting point for each and every new interaction with regard to LT. This reasserts the special role that SEO (search engine optimisation) plays in developing the platform.
- Developers/integrators connected with LT are accustomed to standards introduced by such platforms as GitHub and GitLab. They prefer clear and simple design with only the main attributes stated on the main screen. They also rely heavily on user-generated content, i.e., ratings and reviews left by other users, and on openly available communication with the tool/resource provider.
- Another related point is that developers/integrators would prefer to see each resource/tool in context. Users would like to see all the related or connected items for each resource or tool. For example, if there are two versions of the same dataset, then they should be referenced on the individual description pages of each other. Furthermore, if two tools are usually employed in tow, users would prefer to have them linked. For each tool or resource, users also suggested that it would be useful to see related scientific articles, training, blog posts, etc.
- There is a severe divergence between the motivations of research and non-research professionals, which will need to be addressed in the future. Researchers are generally interested in providing a tool or resource but not its further development. Meanwhile, industry professionals are looking for tools/resources that are regularly updated and supported.
- Other professionals, as well as non-LT industry and non-LT researchers in general, find LT to be a very closed domain with a high barrier of entry. It takes quite a long bootcamp period for them to understand LT and its slang and formulate questions and understand the texts. This is a concern, as it suggests that ELG must take care in the design of its content to be comprehensible and engage non-professionals. This difficulty can be overcome with specifically designed content and training, which need to be planned for the future.
- Related to the previous point, non-LT industry and researchers and other professionals expressed a crucial need for additional content centralised in one place. This content might include such things as articles, independent overviews of tools and services, “for dummies” explanation of terms, and information about events and training.
- All interviewees mentioned existing difficulties with licencing and intellectual property rights. Industry professionals also mentioned existing complications with commercial usage of resources and especially with pricing.

It should be stated that the findings above are not final and will be developed further during the course of the next interviews. However, they present a solid starting point for further investigation.

3 MVP User Journeys

3.1 Representative User Journeys

Based on previously defined user scenarios, we have developed two user journeys necessary for the MVP version. For the next releases, additional user journeys will be created. We understand “user journey” to be the ‘experiences a person has when utilising/interacting with something (typically software)’⁴. For the development

⁴ https://en.wikipedia.org/wiki/User_journey

of the user journeys, we cooperated with a user experience (UX) expert, who helped to define the journeys below. The conclusions presented were based on the interviews, on the partial result of surveys and on internal workshops.

Each user journey describes the step by step interaction of the user with the platform (stages of interaction). For each stage, we have described areas that can be implemented by ELG and areas that are outside of ELG. Also, we have included the questions that a user is asking at each stage of the journey and opportunities for additional improvement. The user journey figures use the colour agenda shown in Figure 39.

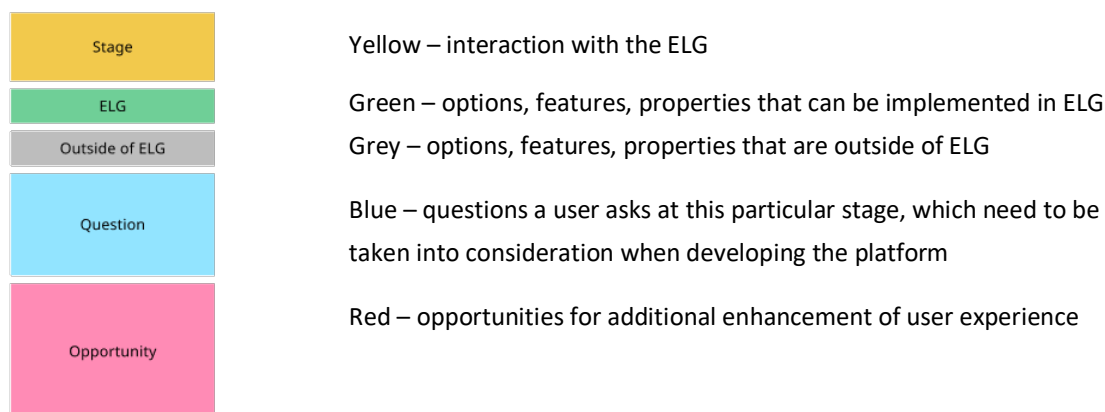


Figure 39. User Journey Agenda

3.1.1 ‘Browsing for the Solution’ User Journey

In this case, the user is not looking for a specific tool. Instead, the user wants to explore what can be found on this website and to evaluate whether it is useful.

In most cases, users will already have some prior knowledge about ELG, either from a newsletter, hearing about ELG from a colleague, learning about it on social media, etc. Such users are already working in the LT field or are involved with the language technology community and, therefore, are familiar with the terminology and what is happening in the industry.

We assume that in this context users might not be looking to solve a specific problem; however, it is possible that they can stumble upon a not yet discovered but useful tool/service/data.

For example, a user might be a developer working in a private LT company on text-to-speech technologies and might be interested in finding valuable datasets. If we provide relevant context on the homepage, they might be willing to explore further and try out a solution offered on ELG.

To accommodate the browsing or exploration scenario, we are introducing a silo page structure, meaning that we create pages for core tool categories that list all relevant content for this particular category. These category pages contain sub-categories that list all the content for the particular sub-category.

Browsing for a solution

LT industry developer or academic



Figure 40. Browsing for a Solution User Journey

3.1.2 'Searching for the Solution' User Journey

In this case, a user has a particular query in mind and wants to see what can be found on ELG. A user might recall a familiar problem and wants to see what solutions are offered on ELG. For example, a user might search using the query "Croatian PoS tagger".

Similar to the browsing scenario, it is safe to assume that the user arrives at ELG with prior knowledge of what to expect and is familiar with LT terminology and the community in general. It is important to note that this context differs from the browsing scenario in that the user might not see value in ELG if the query returns none or non-relevant results. In contrast, during the browsing scenario, users understand the scope of the content by exploring the website and seeing other content, and even though useful content might not be found, such a user might return later in a different context. This user journey presupposes the following effect on the site structure: the results page should contain a faceted search that would allow the user to narrow down the results if the query was too broad or to change the keyword(s) if the query was too narrow. The search page should allow users to evaluate tools without going deeper into the tool landing page.

Searching for a solution

LT industry developer or academic

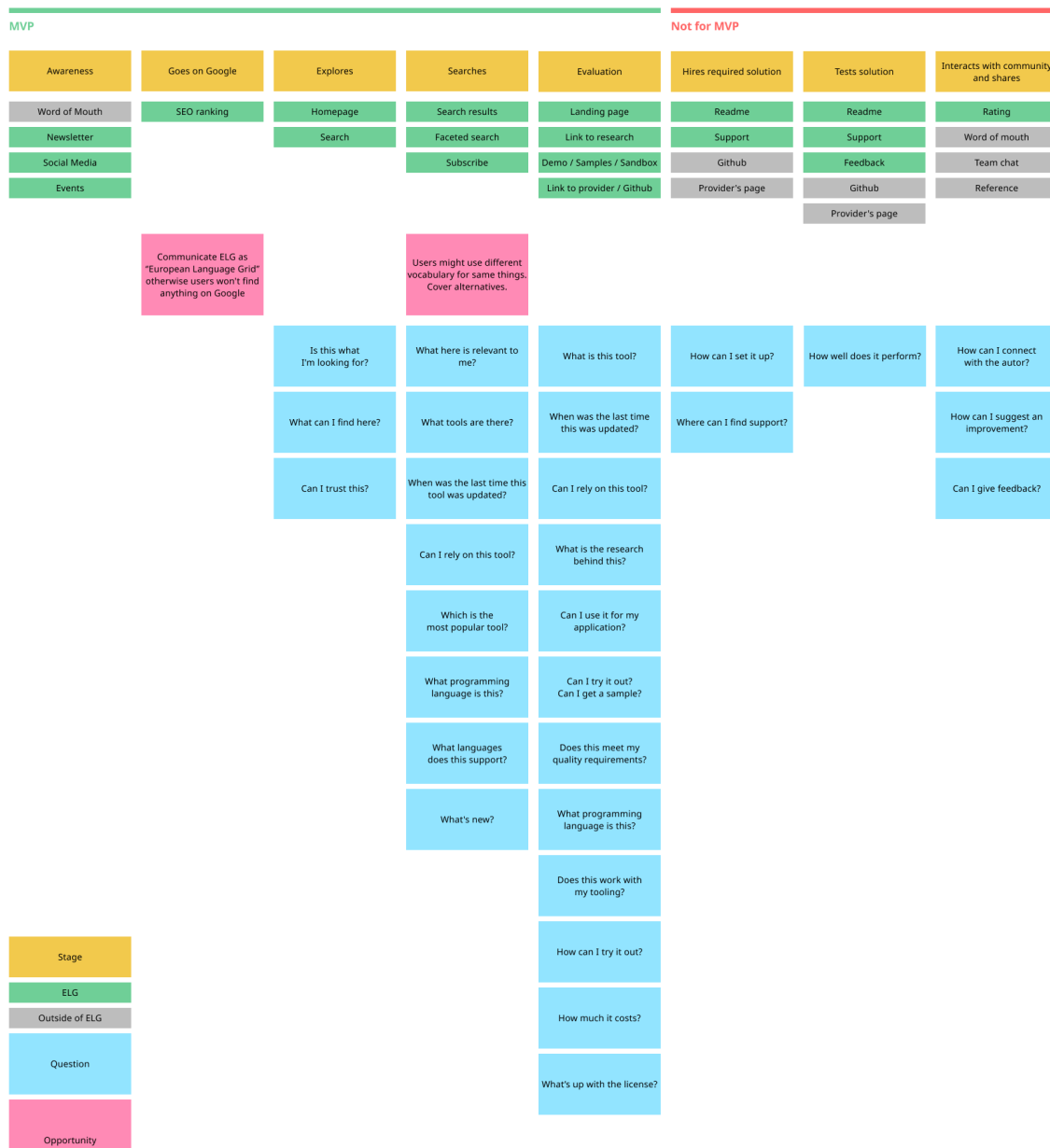


Figure 41 Searching for a Solution User Journey

3.2 Initial site structure

The initial site structure is as follows. Please see the included mock-ups for a more detailed view.

- Homepage
 - Category silo⁵ – Language tools and scripts
 - Sub-Category silo page (for example, Transcribed speech)
 - Sub-Category silo page
 - Sub-Category silo page
 - Results page
 - Resource 1
 - Resource 2
 - Category silo – Language data and Corpora
 - Category silo – Language service providers
 - Search
 - Results page
 - Resource 1
 - Resource 2
 - Faceted search
 - About ELG
 - Subscribe to newsletter

It is important to note that although labels might not represent the accurate academic terminology, these will provide contextual meaning to users not familiar with LT terminology.

3.3 Specific pages

3.3.1 Homepage

For the MVP phase, the homepage will direct users to four paths:

- Search
- Browsing/Exploring through silo pages/landing pages
- Information about the ELG project
- Subscribing to the ELG newsletter

The homepage should provide multiple avenues of exploring ELG content if the user decides to use the menu:

- without interacting with any of the content blocks
- explore the page by scrolling
- using search without delving deeper

During further iterations, different ways of exploring content might be added. The homepage should be attractive not only for new users but should provide value to returning users as well.

⁵ Silo page – page with related content grouped together.

3.3.2 Category Silo Pages

Based on our initial user research, we concluded, that there are three specific user groups interested in three different categories of content (for the MVP version):

- Looking for LT tools and services
- Looking for data and language resources
- Looking for other LT-related information

These groups already illustrate our three core category silo pages. Users might belong to more than one group; therefore, it is important to have solid in-page navigation using breadcrumbs.

Silo pages list all content that falls under a certain category and provide additional context, i.e., describe the category. It is important to provide a general description using common terminology, as this description will help search engines to index and rank the pages better and also help the pages serve as landing pages for short tail keywords, e.g., “corpora”. Therefore, we suggest that commonly used terminology should be included in the category description. For example, for the language data category, we might also include the term “text corpus” somewhere in the description.

Silo pages direct the user towards the next step of narrowing down the content; this is the main purpose of using subcategories. For example, language tools might contain subcategories such as “PoS taggers” or “Segmenters”, that lead the user further into a subcategory silo page.

A single category can hold thousands of tools, and therefore, it is worthwhile to also provide other contexts for browsing (not only subcategories) such as “Most popular language data”, “Recommended tools”, etc. This will make it easier for inexperienced users to navigate the category.

As we broaden our scope by including users that are not experts in the LT field, category silo pages may accommodate additional content beyond descriptions and curated lists. This could include training materials, blog posts, case studies, etc. For example, there could be a featured article titled “Getting started with language tools”. This might also mean expanding the number of categories.

3.3.3 Subcategory Silo Pages

These silo pages differ from the category silo pages with less content and have only subcategory description. They use search result components instead of curated lists to display content. This is done in order to reduce the load of content creation and curation, as such pages are constructed for each subcategory. Users that would end up on subcategory silo pages would probably already be somewhat familiar with language tools and would only need confirmation that they are on the right page.

By using the search results component, we can allow users to further refine results based on their preferences and create many landing pages for search engine optimisation. For example, a user visiting the transcribed speech subcategory under the language data category might want to filter the results by dialect.

As we iterate further, we might create additional landing pages for the most popular long-tail search engine queries, meaning that the user searching for “transcribed speech French” would arrive at a transcribed speech subcategory silo page listing all language data and with an enabled language filter for “French”.

3.3.4 Search Results Page

The search results page lists all the content relevant to the particular query. This page should provide enough information for the user to decide which tools he/she should investigate further.

In the case when the search query is too broad, we allow the user to use faceted navigation to filter the results down further. Faceted navigation differs based on the selected category – for example, language data and tools will have different filtering options. Category tabs, below the search box, allow users to filter content types. In further iterations, it is possible to add other category types, such as articles, case studies, etc.

Only the most popular filtering options will be listed at first; this might include language for language data and programming language for tools. Users might expand filtering options as they continue to narrow down search results.

Since some queries might bring up a lot of content even when results have been narrowed down using filtering options, the user might struggle to find the relevant content without going through large lists. Therefore, we opted for pagination instead of lazy-loading.

As mentioned before, it is important to provide additional context to the user so that he/she can decide to explore further. Therefore, additional data, such as description, rating, date and last update, have been added. By sorting data in different dimensions, users can also sort tools based not only on relevance but also on the best rating, date and last update.

3.3.5 Landing Pages

A landing page describes each category item in detail and allows the user to obtain the particular item. The method differs for each category, e.g., downloading vs. API usage. The same goes for the evaluation process. Before investing in purchasing or subscribing to a certain resource/tool or service, the user might want to try it out first – to download a sample of language data, see a demo of a tool or try out a service.

Users can also see all the necessary attributes for a single content item, such as tags, date when last updated, language, etc. Content providers can also provide a rich text description that lists the benefits, functionality and additional instructions. During further stages, we might include a table of contents, so the provided readme text could be navigated more easily; however, this will depend on how users will interact with this element and what content will be provided by content providers.

Once the user has expressed interest in a particular content item and wants to explore further, they might require additional info – Is there any support? How do I set it up? What is the cost of the license? Such information should also be provided by the content provider and is listed in tabs below the title of the tool. Tabs are used in order to keep the landing page at a decent length, as ELG will have little control over the length of the readme text. This pattern is somewhat similar to what is used on GitHub, and users should already be familiar with it.

3.3.6 Content Page

Content pages are generic pages that contain generic content that does not require additional structure or filtering. During the MVP phase, such pages will contain basic information about the ELG project and might link to other pages.

Such pages might include text/images/videos/links and are possible to edit using a rich text editor. As we iterate further, such pages might be customised for different types of content, for example, blog posts, case studies and training articles.

3.3.7 Information Architecture

Please note that at this stage, we are still dealing with assumptions on how users categorise and request content types. In order to explore this topic, additional user tests should be conducted, for example, card sorting, tree testing, etc. Different new user journeys might be discovered as we investigate further.

3.4 Content guidelines

3.4.1 Search Engine Optimisation

Search engine optimisation (SEO) will greatly improve free, organic traffic coming to the website, especially sending users who are looking for specific solutions or answers, towards ELG. We should specifically keep this in mind when designing the site structure. Although different strategies exist for achieving higher search engine results page rankings, the right strategy (as advised by Google) is to provide as much value to the user as possible. If we want ELG to be used outside of the LT industry, we must keep in mind that we also have to provide value to these users, including secondary use cases:

- researchers who are working in other fields and might find LT useful
- professionals who are working in areas not related to LT and might find LT useful
- developers who are working for a private or governmental institution that is not directly related to LT
- private and governmental organisations looking for a service provider

For these users to find ELG in their search results, it is important to understand that these users might not be familiar with LT terminology, but they are still seeking useful information, tools and resources that ELG might provide.

For example, users outside of the LT industry might be looking for a “machine translation software”, so in order to link these users with LT service providers, we must own valuable content that could later lead such users to contact providers listed on our page. This would, however, mean that we should be able to create content that might explain in simple terms how machine translation works and what types of solutions currently exist in the market.

User interviews, keyword research and prioritisation can provide us with answers about the types of mental models our users have and what keywords they use. By creating content that uses complex academic terminology and does not serve user needs, we risk being outranked by other websites and missing out on significant organic traffic.

It is important to note that users who already have an awareness of ELG and are actively working in LT related industries might still use search engines to find the ELG; therefore, it is important how the name of the platform (and project) is communicated. In cases when we use the abbreviation ‘ELG’ as a keyword, it might not return the right page, as this abbreviation might also stand for other organisations. Therefore, the full name of the platform, “European Language Grid”, should be used when communicating.

3.4.2 Future Content Development

As we continue to design ELG and develop user journeys for users beyond the LT community, reaching such users through engaging content might prove to be a great challenge. In some cases, users will lack the awareness about the possibilities and basic working principles of LT technology. Thus, we do have to further explore ways to engage with such users through relevant content – such content might include case studies, training, explanatory content and even videos.

3.4.3 Keeping in Touch through Newsletters

As we continue to improve ELG and develop user journeys for users beyond the LT community, reaching such users through engaging content might prove to be a great challenge. In some cases, users will lack the awareness about the possibilities and basic working principles of LT technology. Thus, we do have to further explore ways to engage with such users through relevant content – such content might include case studies, training, explanatory content and even videos.

3.5 Sketches

When preparing sketches, we have applied the “mobile first” approach of the initial design for the smallest screen and then work our way up to larger ones. There are several reasons to choose this approach:

- Mobile design is much harder than for desktop screens. By employing this approach, the most difficult questions of UX are resolved in the beginning, essential features are prioritised, and the product is developed in a neat and lean way.
- Mobile internet usage has already surpassed desktop usage. Even for such specialised websites as the ELG platform, the number of mobile visitors will be very high. This is especially important for first impression users, those who have just heard about ELG and try to open it on their mobile devices.
- Google ranks websites that have mobile versions higher, which is essential for ELG.

3.5.1 Mobile Sketches

Presented below are the sketches for mobile devices developed according to the outputs from the user journeys.

From: elg@elg.com
To: janis@tilde.lv
Subject: European language grid is launched.



European Language Grid is now launched

All European language tools, data and
services in one place.

Visit ELG platform


Figure 42. Mobile Newsletter



European Language Grid


All European language tools, data and services in one place.

Explore by topic




Language technologies

Scripts, that include tools for translation, machine learning, language processing, speech recognition.



Language data and resources

Data, that can be used to train machine learning algorithms. This can include speech transcriptions, multilingual translations.



LT related information

Learn more about Language technologies. Get relevant training

What are language technologies?

Language technology, often called human language technology (HLT), studies methods of how computer programs or electronic devices can analyze, produce, modify or respond to human texts and speech. It consists of natural language processing (NLP) and computational linguistics (CL) on the one hand, and speech technology on the other. It also includes many application oriented aspects of these. Working with language technology often requires broad knowledge not only about linguistics but also about computer science.

[Learn more about technology](#)

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

Figure 43. Mobile Homepage



Search

All [Technologies](#) Resources Information

324 results

PoS Taggers X latvian X

Extractor

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Segmenter

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago



Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Figure 44. Mobile Search results page



Part-of-speech taggers

Search & Filter

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

324 results

PoS taggers X

Extractor

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Segmenter

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines



1 updated 2 months ago

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

Figure 46. Mobile Silo secondary page



Explore language tools and scripts

Search & Filter

What are language tools? Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Top Categories

Spell checkers

Speech recognition

POS Taggers

Segmenters

Tokenizers

Extractors

Parsers

Commatizer

Show all

Recently Updated

Segmenter

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Show more

Most Popular

Segmenter

2006 CoNLL Shared Task - Ten Languages

Tool description in a couple of words not more than a two lines

46 updated 2 months ago

Speech Recognition

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Tokenizer

Aalto University Automatic

Tool description in a couple of words not more than a two lines

1 updated 2 months ago

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

Figure 45. Mobile Silo page

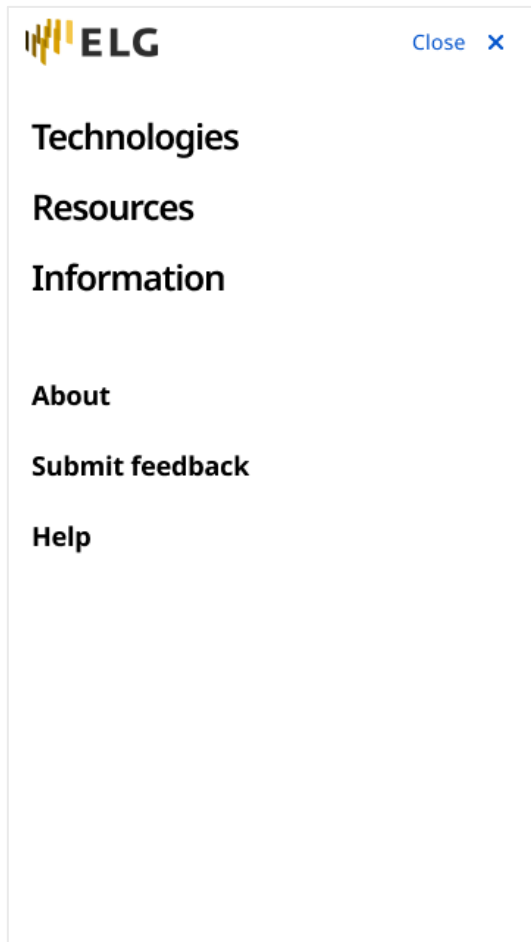


Figure 48. Mobile Menu page

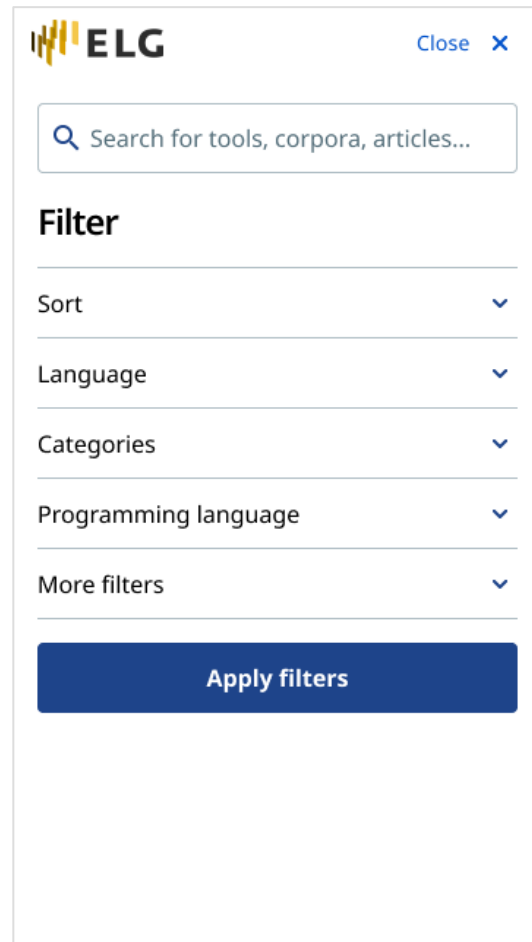
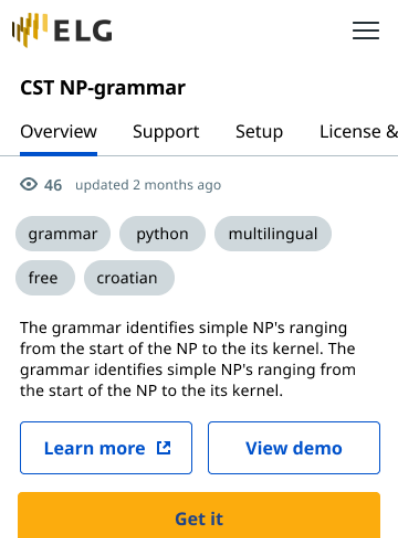


Figure 47. Mobile Filter page



The mockup shows a mobile interface for the 'CST NP-grammar' item. At the top is the ELG logo and a hamburger menu icon. Below is the title 'CST NP-grammar' and a navigation bar with 'Overview', 'Support', 'Setup', and 'License &'. The 'Overview' tab is active. Below the navigation bar, it shows '46' views and 'updated 2 months ago'. There are five tags: 'grammar', 'python', 'multilingual', 'free', and 'croatian'. A description follows: 'The grammar identifies simple NP's ranging from the start of the NP to the its kernel. The grammar identifies simple NP's ranging from the start of the NP to the its kernel.' Below the description are two buttons: 'Learn more' with an external link icon and 'View demo'. At the bottom is a large orange button labeled 'Get it'.

Heading 1

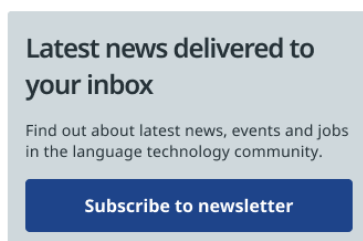
In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Heading 2

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Heading 3

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.



A light blue box with the text 'Latest news delivered to your inbox' in bold. Below it, in smaller text, is 'Find out about latest news, events and jobs in the language technology community.' At the bottom is a dark blue button with the text 'Subscribe to newsletter' in white.

Figure 49. Mobile Item page

3.5.2 Desktop Sketches

Presented below are the sketches for desktop devices developed according to outputs from the user journeys.

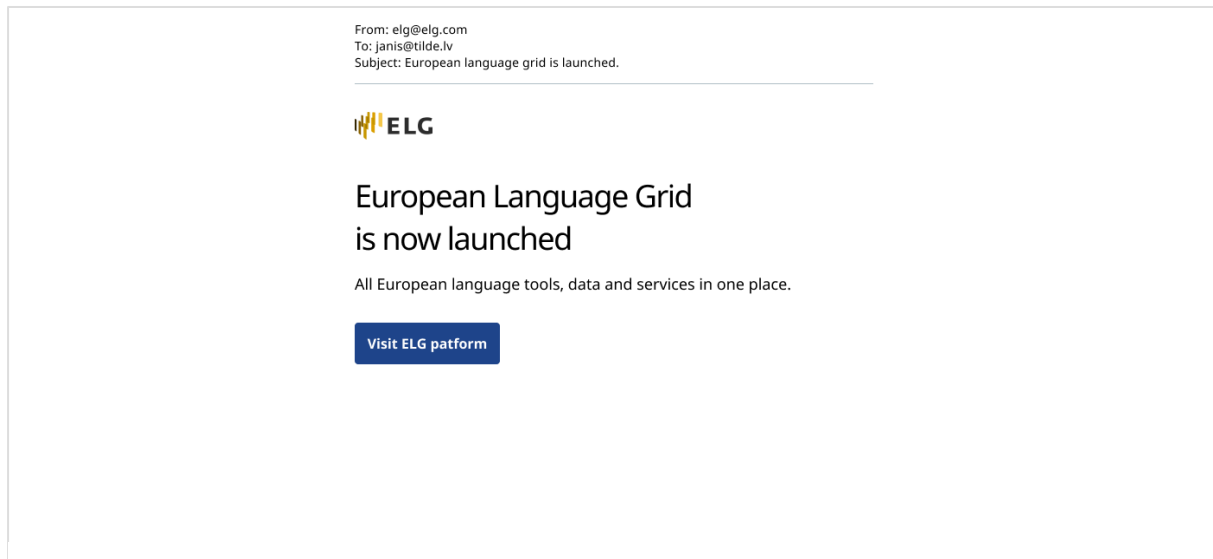


Figure 50. Desktop Newsletter




European Language Grid

All European language technologies, data and information in one place.


Search

Explore by topic




Language technologies

Scripts, that include tools for translation, machine learning, language processing, speech recognition.



Language data and resources

Data, that can be used to train machine learning algorithms. This can include speech transcriptions, multilingual translations.



LT related information

Learn more about Language technologies. Get relevant training

What are language technologies?

Language technology, often called human language technology (HLT), studies methods of how computer programs or electronic devices can analyze, produce, modify or respond to human texts and speech. It consists of natural language processing (NLP) and computational linguistics (CL) on the one hand, and speech technology on the other. It also includes many application oriented aspects of these. Working with language technology often requires broad knowledge not only about linguistics but also about computer science.

[Learn more about technology](#)

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

Figure 51. Desktop Homepage

All **Technologies** Resources Information


Categories	324 results	Sort: Best match ▾
Speech Recognition (23)	PoS taggers ✕ latvian ✕	
Tokenizer (19)		
Spell checkers (8)	CST NP-grammar	Tokenizer 24
Tokenizer (1)	Tool description in a couple of words not more than a two lines updated 2 months ago	
Commatizer (1)		
Show all	Aalto University Automatic	Tokenizer 24
Language	Tool description in a couple of words not more than a two lines updated 2 months ago	
French (23)	2006 CoNLL Shared Task - Ten Languages	Commatizer 1
Latvian (19)	Tool description in a couple of words not more than a two lines updated 2 months ago	
Italian (8)	88miSMS. A corpus of authentic text messages in French	Tokenizer
Croatian (1)	Tool description in a couple of words not more than a two lines updated 2 months ago	
Multilingual (1)		
Programming language	Aalto University Automatic	Spell checkers 24
Python (23)	Tool description in a couple of words not more than a two lines updated 2 months ago	
TeX (1)	2006 CoNLL Shared Task - Ten Languages	Tokenizer 24
Java (1)	Tool description in a couple of words not more than a two lines updated 2 months ago	
License type	88miSMS. A corpus of authentic text messages in French	Segmenter 24
Free	Tool description in a couple of words not more than a two lines updated 2 months ago	
Commercial use	2006 CoNLL Shared Task - Ten Languages	Speech Recognition 35
Creative commons	Tool description in a couple of words not more than a two lines updated 2 months ago	
More filters...	Aalto University Automatic	Tokenizer 24
	Tool description in a couple of words not more than a two lines updated 2 months ago	
	2006 CoNLL Shared Task - Ten Languages	Commatizer 1
	Tool description in a couple of words not more than a two lines updated 2 months ago	

Previous 1 2 **3** 4 5 ... 39 40 Next

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Figure 52. Desktop Search page



[Home](#) [Technologies](#) [Resources](#) [Information](#) [About ELG](#)

[Home](#) > [Tools](#)

Explore language technologies

What are language tools? Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Q

Search for language technologies

Search

Top Categories

[Spell checkers](#)
[Speech recognition](#)
[POS Taggers](#)
[Show all](#)

[Segmenters](#)
[Tokenizers](#)
[Extractors](#)

[Commatizer](#)
[Speech recognition](#)
[POS Taggers](#)

[Spell checkers](#)
[Speech recognition](#)
[POS Taggers](#)

Recently updated

[Segmenter](#)
CST NP-grammar
Tool description in a couple of words not more than a two lines

46

updated 2 months ago

[Speech Recognition](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Tokenizer](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Show more](#)

[Speech Recognition](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Tokenizer](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Tokenizer](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Show more](#)

Most Popular

[Segmenter](#)
2006 CoNLL Shared Task - Ten Languages
Tool description in a couple of words not more than a two lines

46

updated 2 months ago

[Speech Recognition](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Tokenizer](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Show more](#)

[Segmenter](#)
2006 CoNLL Shared Task - Ten Languages
Tool description in a couple of words not more than a two lines

46

updated 2 months ago

[Tokenizer](#)
Aalto University Automatic
Tool description in a couple of words not more than a two lines

1

updated 2 months ago

[Segmenter](#)
2006 CoNLL Shared Task - Ten Languages
Tool description in a couple of words not more than a two lines

46

updated 2 months ago

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

[Home](#) [Technologies](#) [Resources](#) [Information](#) [About ELG](#)

[Submit feedback](#) [Help](#)

©2019 ELG Consortium [Terms of Use](#)

Figure 53. Desktop Silo page

ELG

50/103

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Previous 1 2 **3** 4 5 ... 39 40 Next

Find out about latest news, events and jobs in the language technology community.

Subscribe to newsletter

ELG



CST NP-grammar

[Overview](#) [Support](#) [Setup](#) [License & Pricing](#)

updated 2 months ago by [Tilde SIA](#) [Contact provider](#) [24](#)

[grammar](#) [python](#) [multilingual](#) [free](#) [croatian](#)

The grammar identifies simple NP's ranging from the start of the NP to its kernel. The grammar identifies simple NP's ranging from the start of the NP to its kernel.

[Get it](#)

[View demo](#)

[Learn more](#)

Heading 1

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Heading 2

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Heading 3

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

Latest news delivered to your inbox

Find out about latest news, events and jobs in the language technology community.

[Subscribe to newsletter](#)

Figure 55. Desktop Item page

3.6 Accessibility

By following the standardised W3C WCAG 2.0 accessibility guidelines and best practices⁶, we can ensure that a solid foundation for good design, which serves human needs, is established.

Understanding that different users will be visiting our site is of great importance; this not only relates to people with disabilities but to all of our users. We must invest time to become familiar with our end users, so the platform truly serves all our visitors (not limited to LT professionals). Therefore, the platform should also appeal to the broader public interested in LT.

- Using correct contrast ratios
- Using correct visual contrasts to identify the content hierarchy
- Using plain language for written texts
- Using appropriate HTML tags and ensuring text is machine readable
- Providing feedback on the system status
- Not using colour to communicate meaning

⁶ <https://www.w3.org/WAI/standards-guidelines/wcag/>

- Describing the content of images
- Providing text alternatives for non-text content
- Making controls easy to interact with
- Using semantic markup
- Providing clear instructions
- Providing clear visual cues for navigation
- Supporting browser customisation, such as increased text size and high contrast
- Using clean and easy to read typography
- Using audience appropriate language
- Allowing content to be easily scanned using appropriate visual contrasts
- Supporting beginner and expert users
- Give instructions at the right time in the right place
- Include people with disabilities in usability work
- Content appears and operates in predictable ways
- Users are helped to avoid and correct mistakes
- Content is compatible with current and future user tools
- Relying on recognition instead of recall
- Provide help and documentation
- The user interface must comply with usability heuristics

4 Design Guidelines

4.1 Material Design

For the MVP, the Google Material Design⁷ framework was used as a solid foundation on which to build upon. Google Material Design already includes all the necessary web components that can be adapted for the ELG identity and works well together with the Angular front-end framework that is used for building the platform. By using this design system, we can allocate our resources better towards improving the usability of the platform instead of focusing on small details and interactions. Material design is already widely adopted in the industry.

Material Design includes not only visual guidelines for the most common web elements but also takes care of interactions, animations, iconography and colour and has great adaptability for different devices.

With the recent introduction of Material Theming, we no longer have to rely on the stock Material Design look, as all the elements, including typography, colour scheme and appearance of elements can be customised using ELG identity elements.

For a start – we took the foundational parts of the ELG initiative identity defined in D8.1 “ELG Project and Initiative Identity” and created our own Material theme, which includes the defined colour scheme and typography.

The ELG identity includes, among others, two main colours – Council of Europe blue (hex code #1e448a) and a complementary orange that was derived from the colours of the META-NET initiative (hex code #d6ad5c). We

⁷ <https://material.io/design/>

added base colours, which consist of different shades of black, for text, borders and backgrounds by checking the required contrast ratios that are at least 4.5 and thus meeting WCAG 2 level AA accessibility standard.

For the typography, we started by using the predefined typography scale as defined by Material design guidelines – it should give us a good starting point in finding the right text sizes to ensure proper visual contrast and proportions. According to the ELG identity, we adapted Noto Sans as the primary typeface. Noto Sans supports a large array of languages, and this will allow us to display any multilingual data on our platform. Using this scale as a base, we have a solid groundwork for meeting accessibility guidelines and ensuring that the text is readable.

Please note that as we delve further into developing the actual solutions and if we feel that the Theme is forcing us to use an inferior solution, we will have to adjust the material components accordingly.

4.2 Design System and Interface Components

The exported style guide with various elements is presented below. The .scss files will be uploaded to the project's GitLab and will be available to all partners willing to create their own interface for their tools/resources.

The following Material Foundations are included:

- Colour Scheme
- System Icons
- Theme UI
- Typography Scale

The following components are presented (all explanation according to Material Design guidelines⁸):

- Backdrop – appears behind all other surfaces in an app, displaying contextual and actionable content
- Banners – displays a prominent message and related optional actions
- Buttons – allow users to take actions and make choices with a single tap
- Cards – contain content and actions about a single subject
- Chips – compact elements that represent an input, attribute or action
- Dialogs – inform users about a task and can contain critical information, require decisions or involve multiple tasks
- FAB (Floating Action Button) – represents the primary action of a screen
- Lists – continuous, vertical indexes of text or images
- Menus – display a list of choices on temporary surfaces
- Navigation Drawers – provide access to destinations in the app
- Progress Indicators – express an unspecified wait time or display the length of a process
- Selection Controls – allow the user to select options
- Sliders – allow users to make selections from a range of values
- Snackbar – provides brief messages about app processes at the bottom of the screen
- Tabs – organise content across different screens, data sets and other interactions

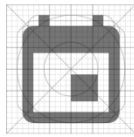
⁸ <https://material.io/design/components/>

- Text Fields – let users enter and edit text



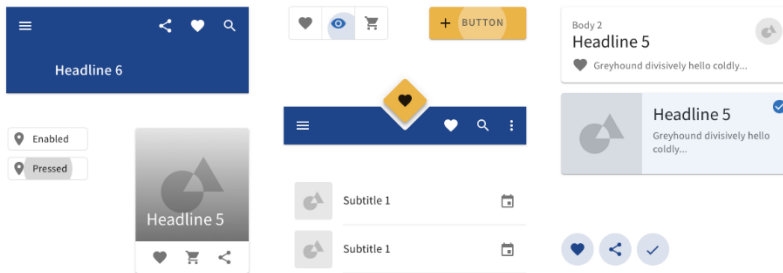
Figure 56. Colour Scheme

System Icons



By default this theme uses the Filled system icon set. Override any icon in this file for a different preset style – Rounded, Sharp, Outlined or Two-toned – by using Overrides in Sketch.

Download full icon sets with Theme Editor and select in the Sketch menu "File > Add as Library" or visit [Material.io/icons](https://material.io/icons)



Filled System Icon Set



Sharp System Icon Set



Rounded System Icon Set



Two-toned System Icon Set



Outlined System Icon Set



Figure 57. System Icons

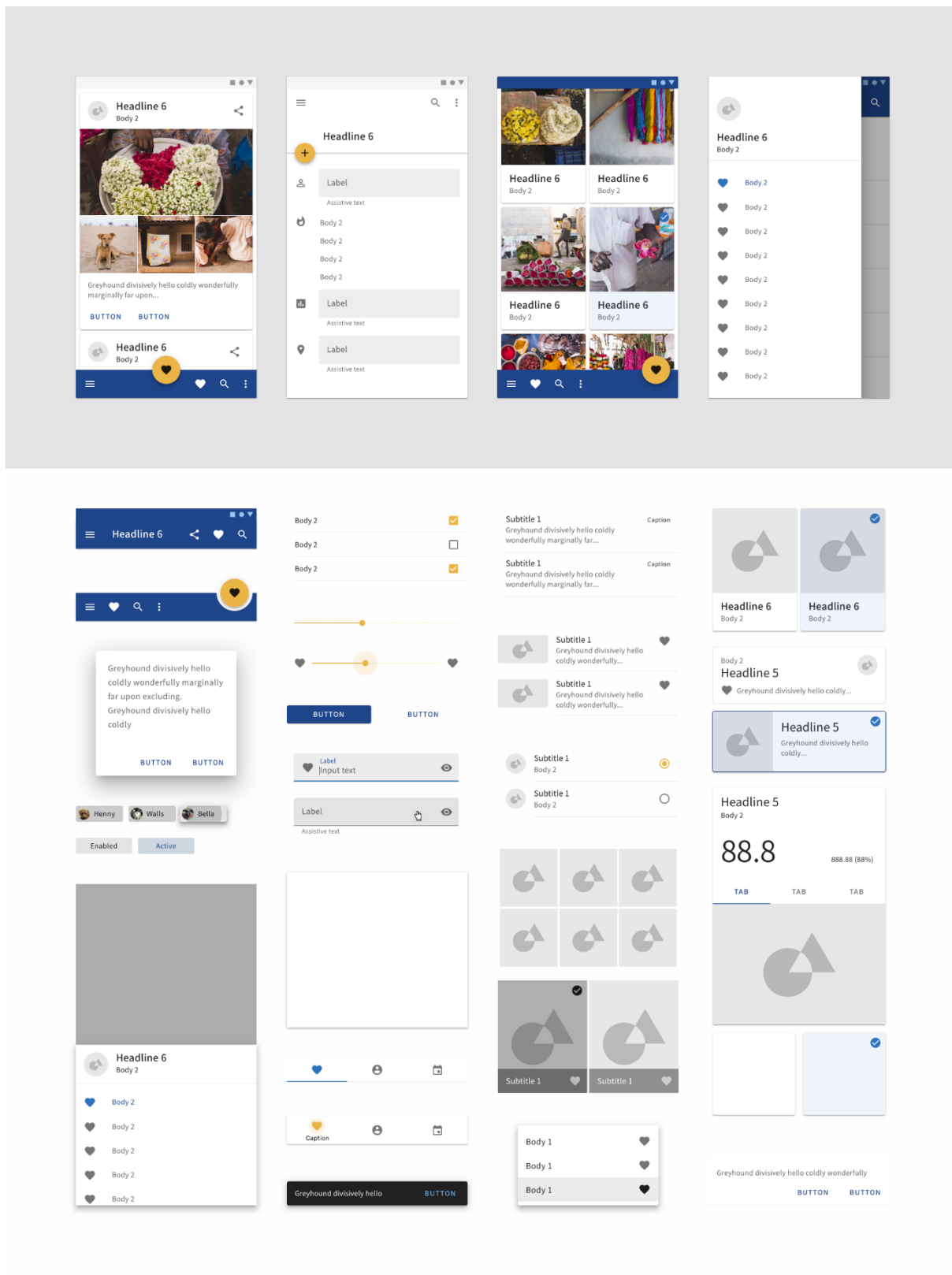


Figure 58. Theme UI

Typographic Scale

Hello, World.
An Adaptable Foundation

OVERLINE
Headline 6
Apparently we had reached a great height in the atmosphere, for the sky was a dead black, and the stars had ceased to twinkle. By the same illusion which lifts the horizon of the sea to the level of the spectator on a hillside, the sable cloud beneath was dished out, and the car seemed to float in the middle of an immense dark sphere, whose upper half...

Headline 4
Headline 5
By the same illusion which lifts the horizon of the sea to the level of the spectator on a hillside, the sable cloud beneath was dished out, and the car seemed to float in the middle of an immense dark sphere, whose upper half was...

Body 1
Caption
By the same illusion which lifts the horizon of the sea to the level of the spectator on a hillside, the sable cloud beneath was dished out...

H1 / Noto Sans CJK KR DemiLig

93.4 / -1.5 letter-spacing

H2 / Noto Sans CJK KR DemiLight

58.38 / -0.5 letter-spacing

H3 / Noto Sans CJK KR Regular

46.7 / 0 letter-spacing

H4 / Noto Sans CJK KR Regular

33.08 / 0.25 letter-spacing

H5 / Noto Sans CJK KR Regular

23.35 / 0 letter-spacing

H6 / Noto Sans CJK KR Medium

19.46 / 0.25 letter-spacing

Body 1 / Noto Sans CJK KR Regular

15.57 / 0.5 letter-spacing

Body 2 / Noto Sans CJK KR Regular

13.62 / 0.25 letter-spacing

Subtitle 1 / Noto Sans CJK KR Regular

15.57 / 0.15 letter-spacing

Subtitle 2 / Noto Sans CJK KR Medium

13.62 / 0.1 letter-spacing

BUTTON / NOTO SANS CJK KR MEDIUM

13.62 / 1.25 letter-spacing

Caption / Noto Sans CJK KR Regular

11.68 / 0.4 letter-spacing

OVERLINE / NOTO SANS CJK KR MEDIUM

11.68 / 2 letter-spacing

Figure 59. Typography Scale

Backdrop

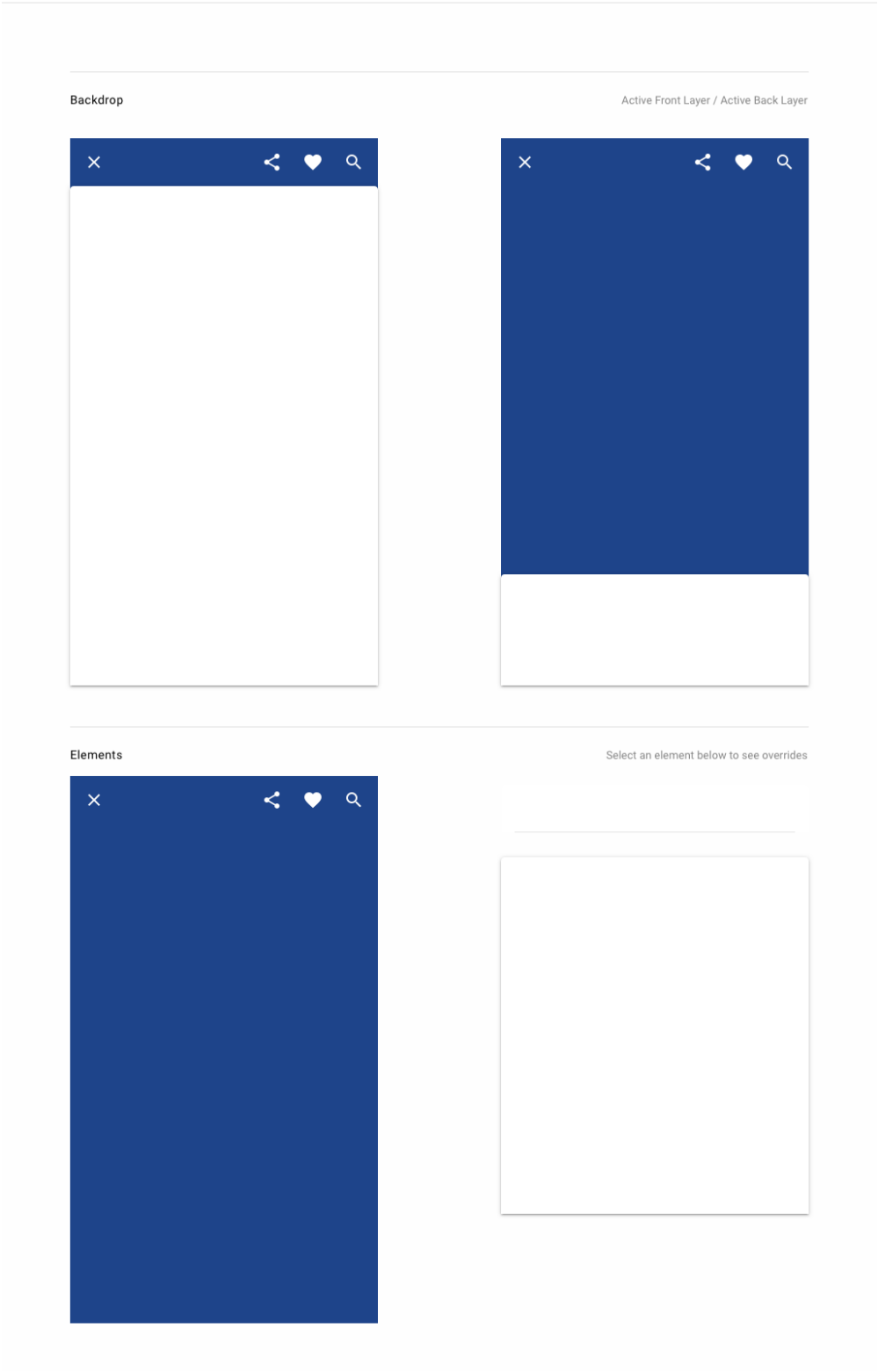


Figure 60. Backdrop

Banner

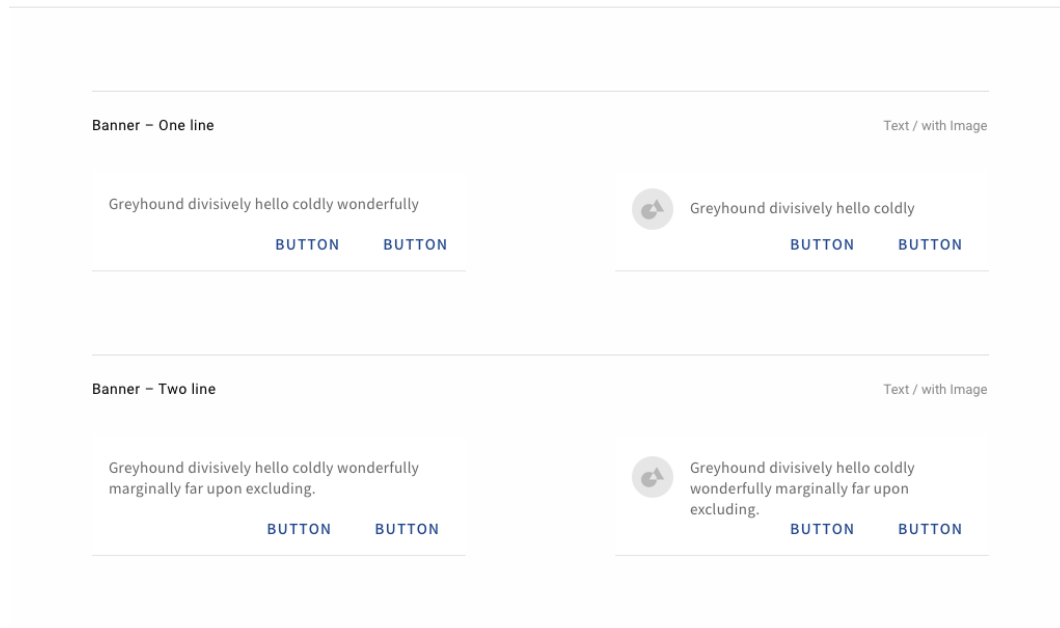
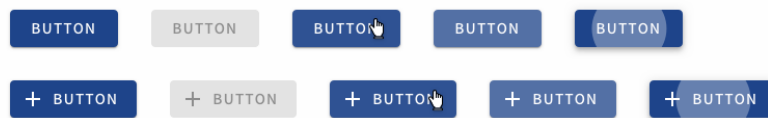


Figure 61. Banners

Buttons

Contained Button

States



Text Button

States



Outlined Button

States



Toggle Button

States & Elements

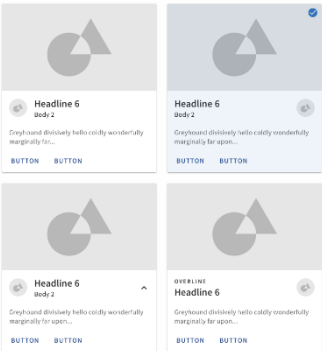


Figure 62. Buttons

Cards

Material.io/Guidelines/Components
Version: 5.24

Card Layouts - Image Top



Card layouts

Card Layouts - Full Image

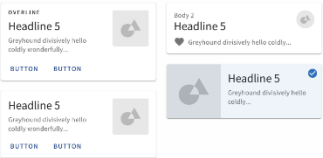


Card layouts with actions

Cards

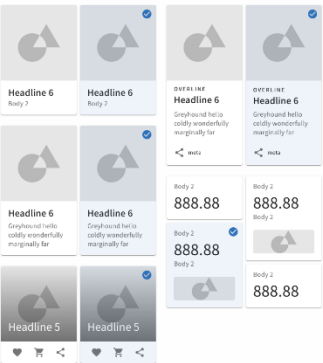
Material.io/Guidelines/Components
Version: 5.24

List 5-up



Layouts

List 5-up



Layouts

Cards

Material.io/Guidelines/Components
Version: 5.24

Card Layouts

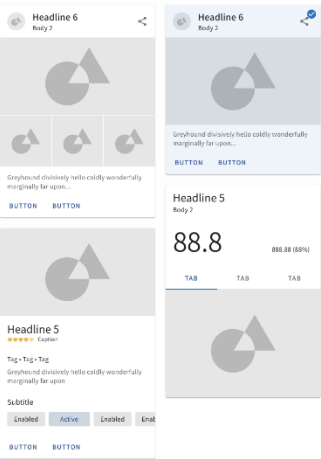


Figure 63. Cards

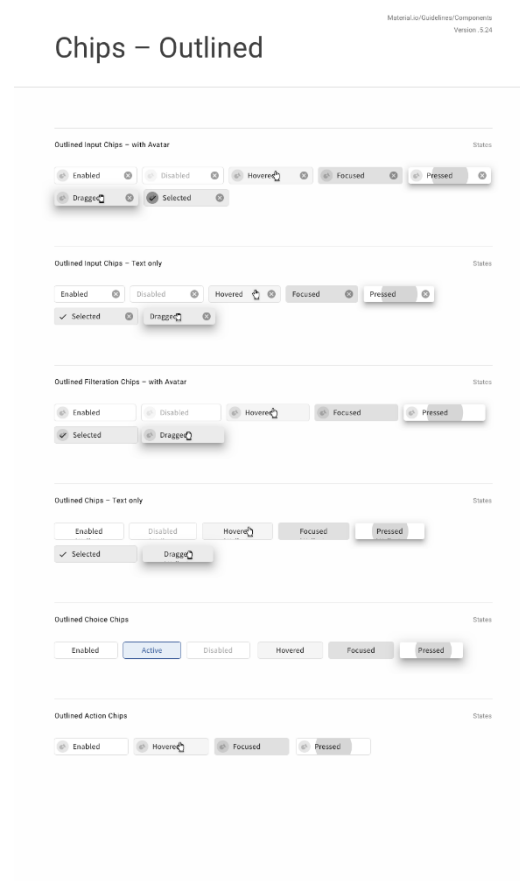
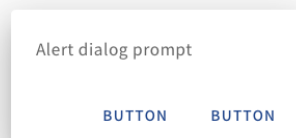


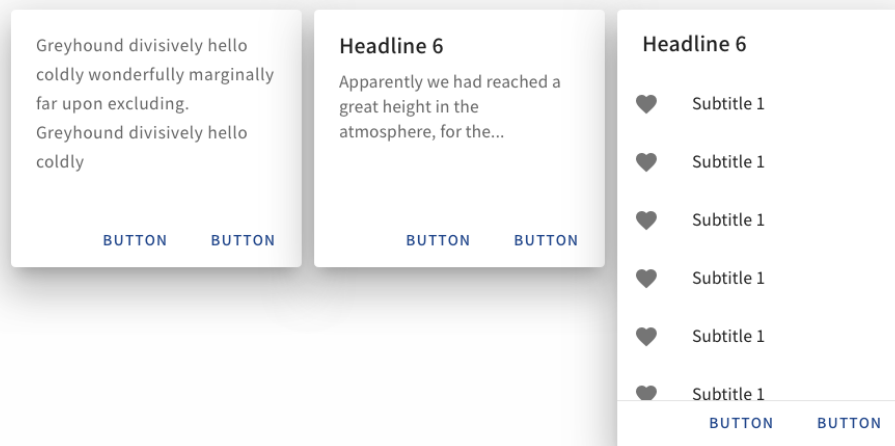
Figure 64. Chips

Dialogs

Alert Dialogs



Confirmation Dialogs



Simple Dialogs

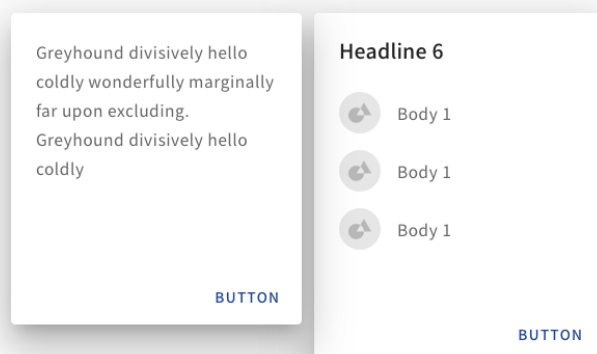


Figure 65. Dialogs

FAB

Standard



States for Primary / Secondary



Mini FAB



States for Primary / Secondary

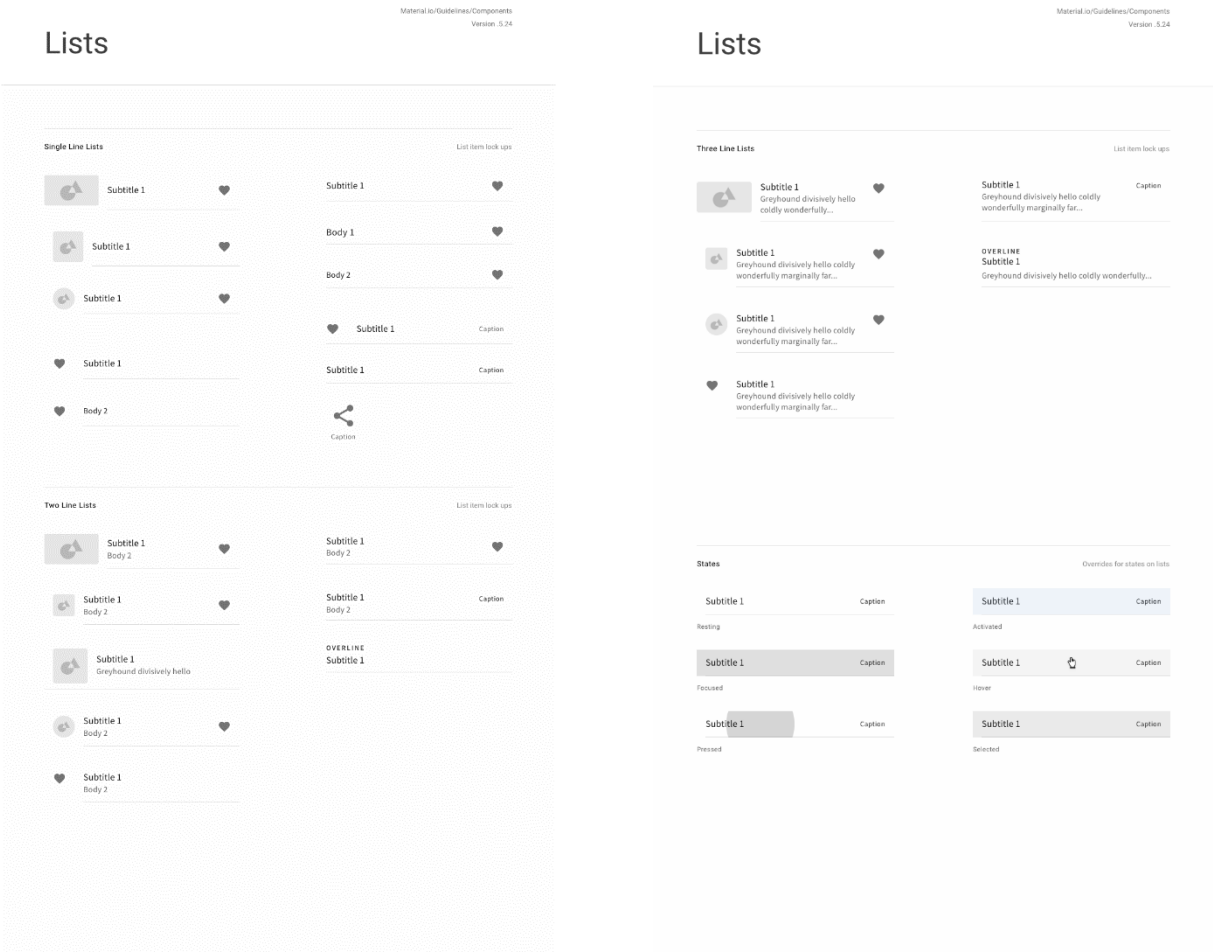


Extended FAB



States for Primary / Secondary

Figure 66. FAB



Material.io/Guidelines/Components
Version: 5.24

Lists

Three Line Lists

List item look-ups

 Subtitle 1

Greyhound divisively hello coldly wonderfully...

Heart icon

Subtitle 1

Greyhound divisively hello coldly wonderfully marginally far...

Caption

 Subtitle 1

Greyhound divisively hello coldly wonderfully marginally far...

Heart icon

OVERLINE

Subtitle 1

Greyhound divisively hello coldly wonderfully...

 Subtitle 1

Greyhound divisively hello coldly wonderfully marginally far...

Heart icon

 Subtitle 1

Greyhound divisively hello coldly wonderfully marginally far...

Heart icon

Heart icon

Subtitle 1

Greyhound divisively hello coldly wonderfully marginally far...

States

Overrides for states on lists

Resting

Subtitle 1

Caption

Activated

Subtitle 1

Caption

Focused

Subtitle 1

Caption

Hover

Subtitle 1

Caption

Pressed

Subtitle 1

Caption

Selected

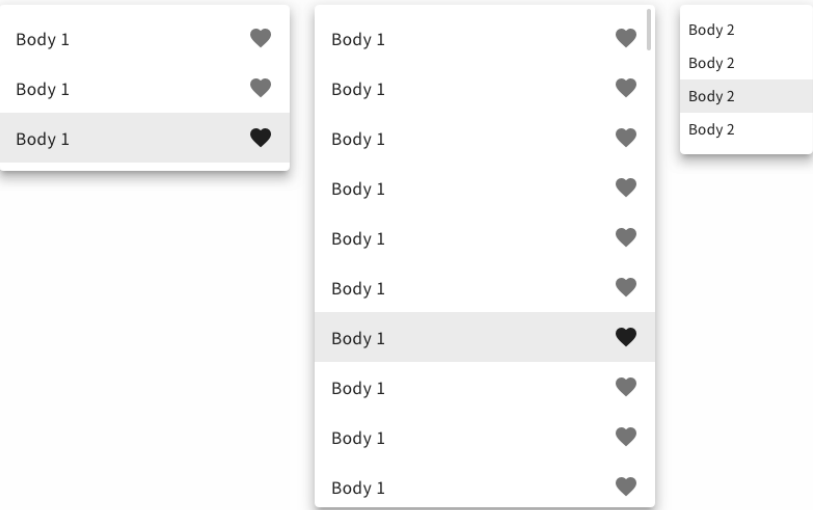
Subtitle 1

Caption

Figure 67. Lists

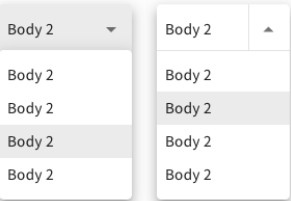
Menus

Dropdown Menus



Exposed Dropdown Menus

Above / Coplanar



Elements

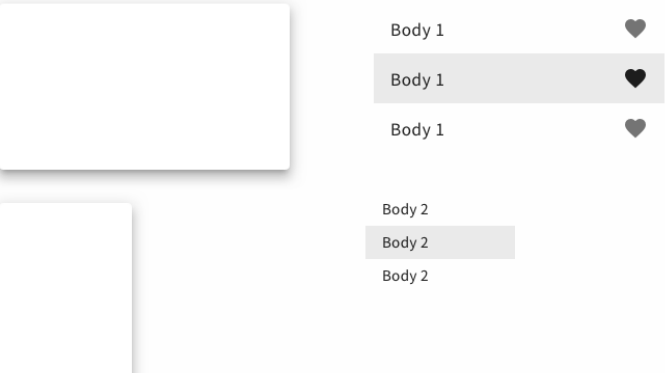


Figure 68. Menus

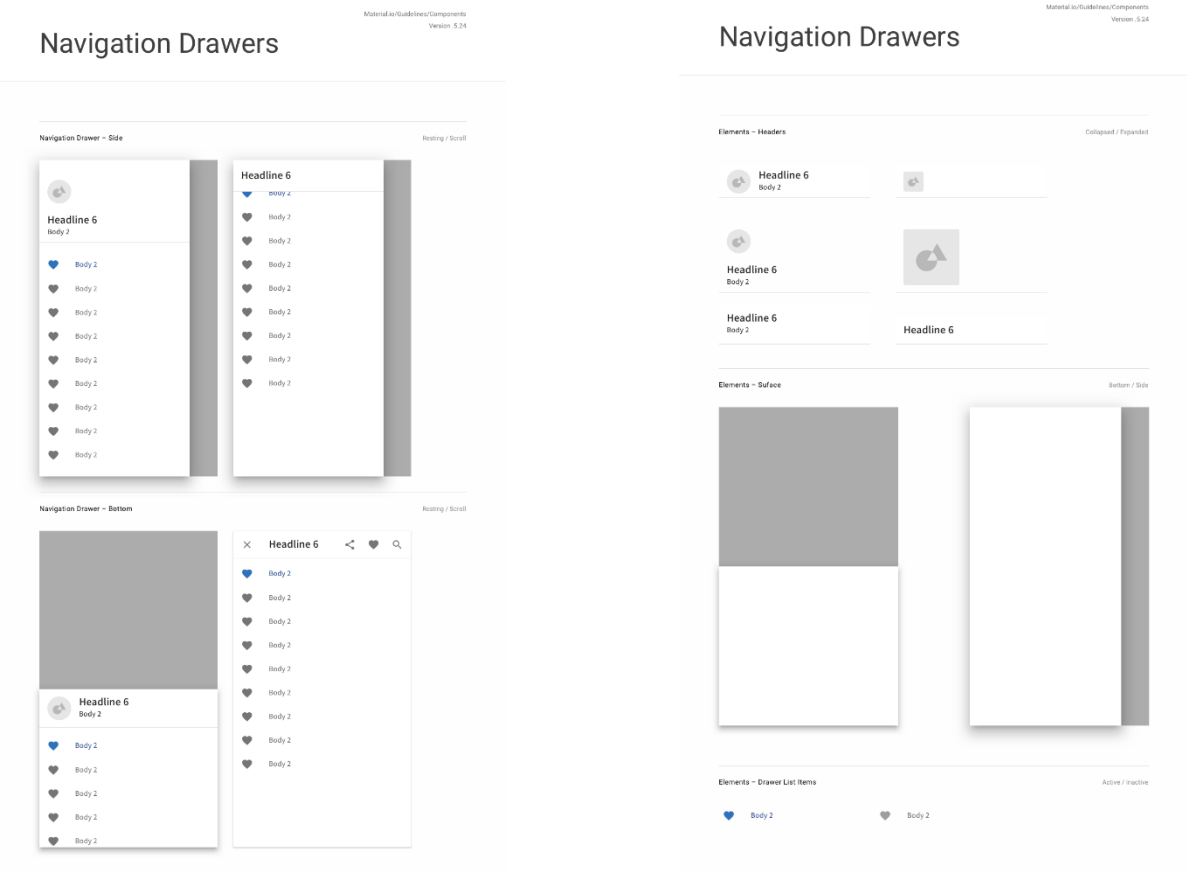


Figure 69. Navigation Drawers

Progress Indicators

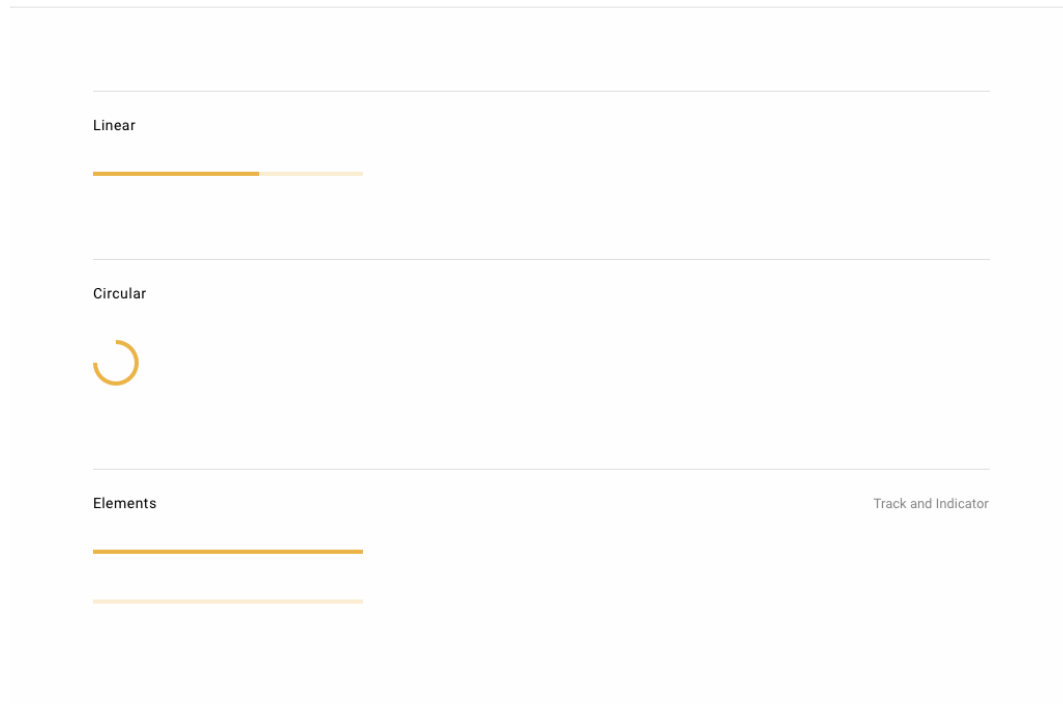


Figure 70. Progress Indicators

Selection Controls

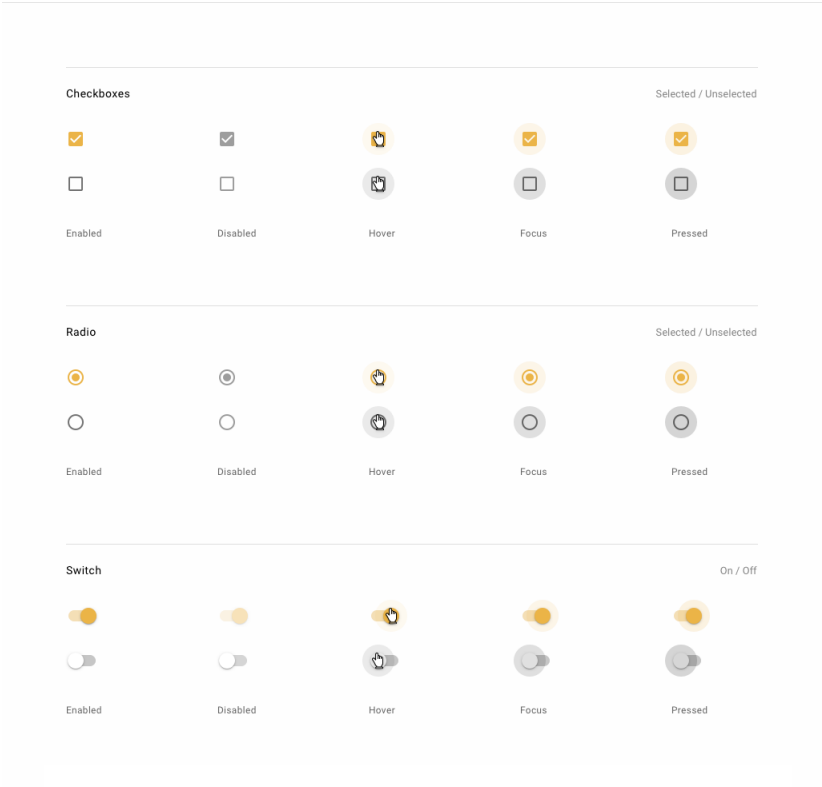


Figure 72. Selection Controls

Snackbar

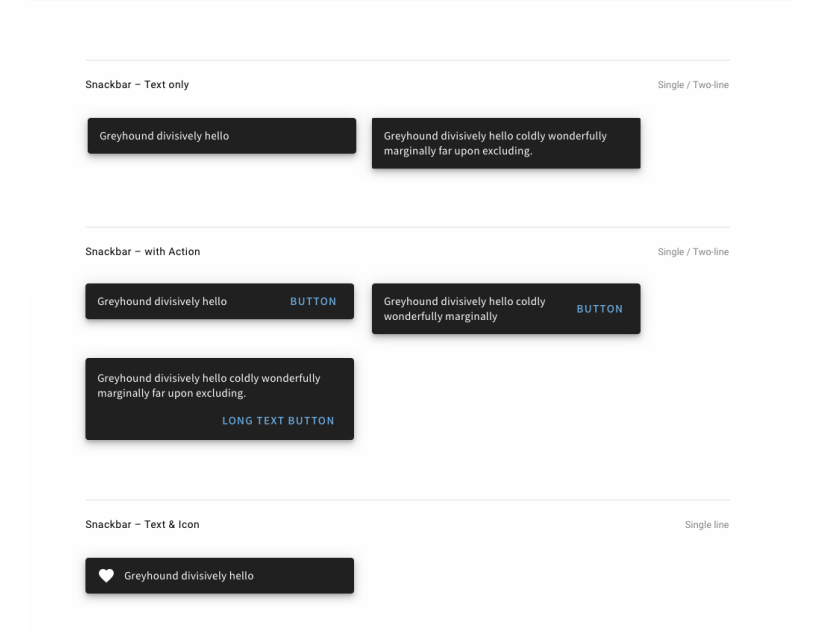


Figure 71. Snackbar

Sliders

Discret

States



Continuous

States



Continuous – Lock Ups

Use overrides to insert a slider states into these locks ups



Discret – Lock Ups

Use overrides to insert a slider states into these locks ups



Range

Standard / with Values



Range – Lock Ups

Use overrides to insert a slider states into these locks ups



Figure 73. Sliders

Tabs

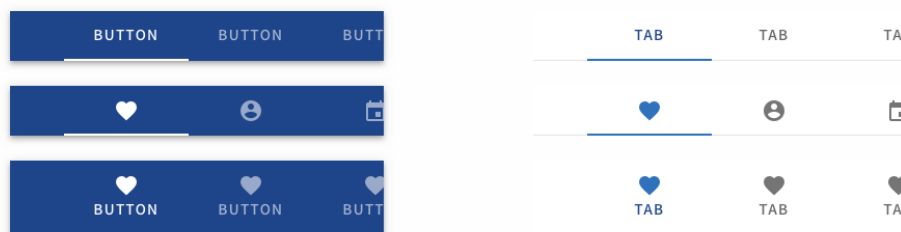
Fixed Tabs

Raised / Flat



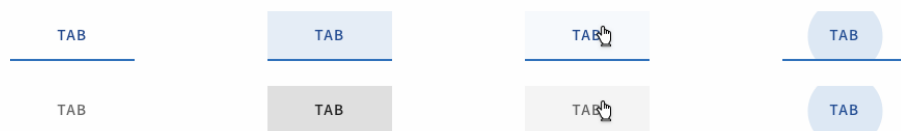
Scrollable Tabs

Raised / Flat



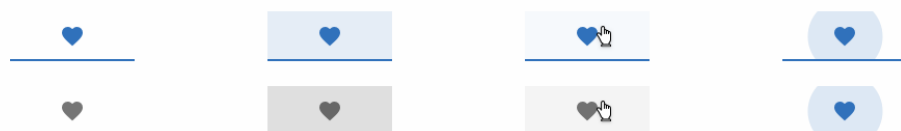
Elements – Text Tabs

Active and Inactive Tab states



Elements – Icon Tabs

Active and Inactive Tab states



Elements – Text & Icon Tabs

Active and Inactive Tab states

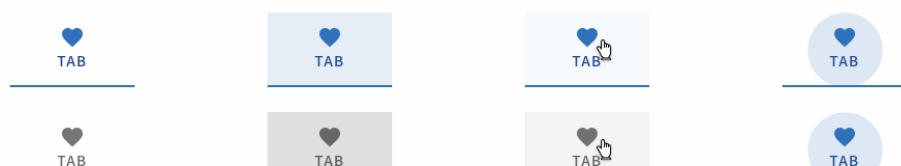


Figure 74. Tabs

Text Fields

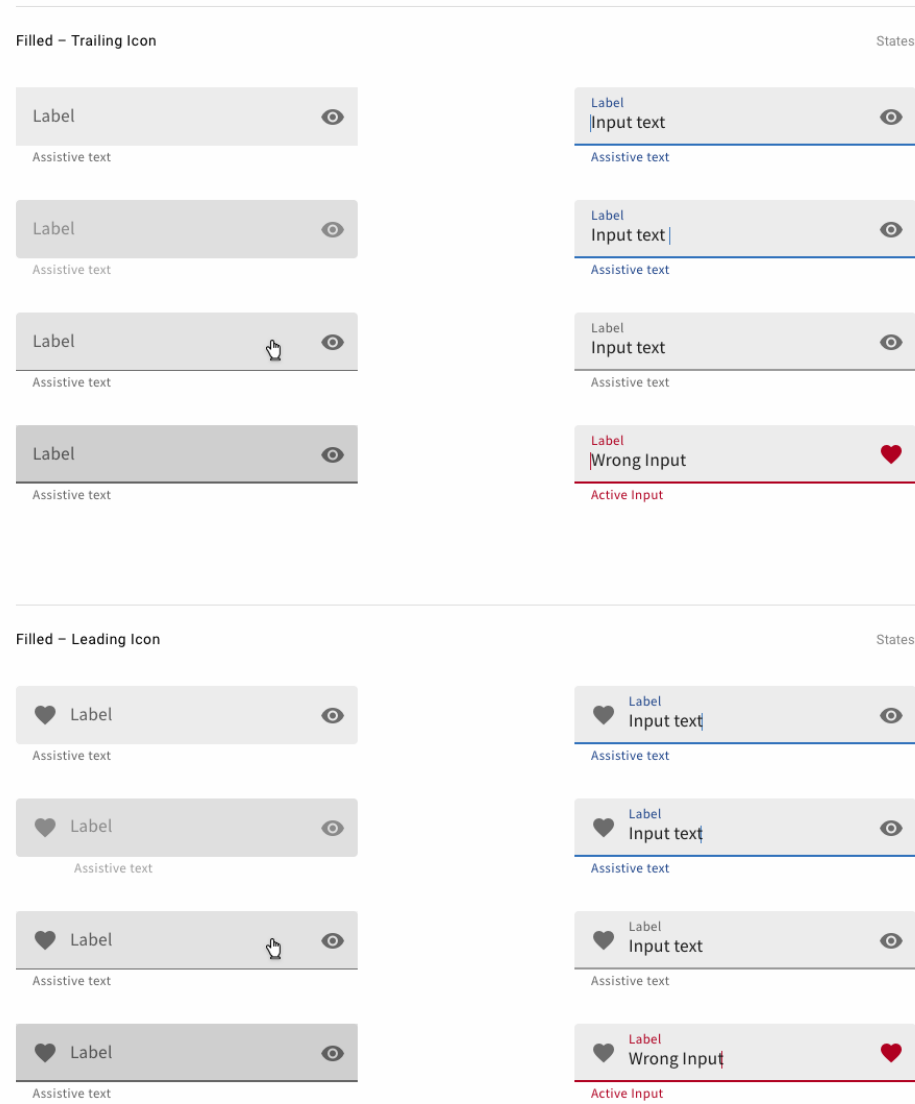


Figure 75. Text Fields

Text Fields

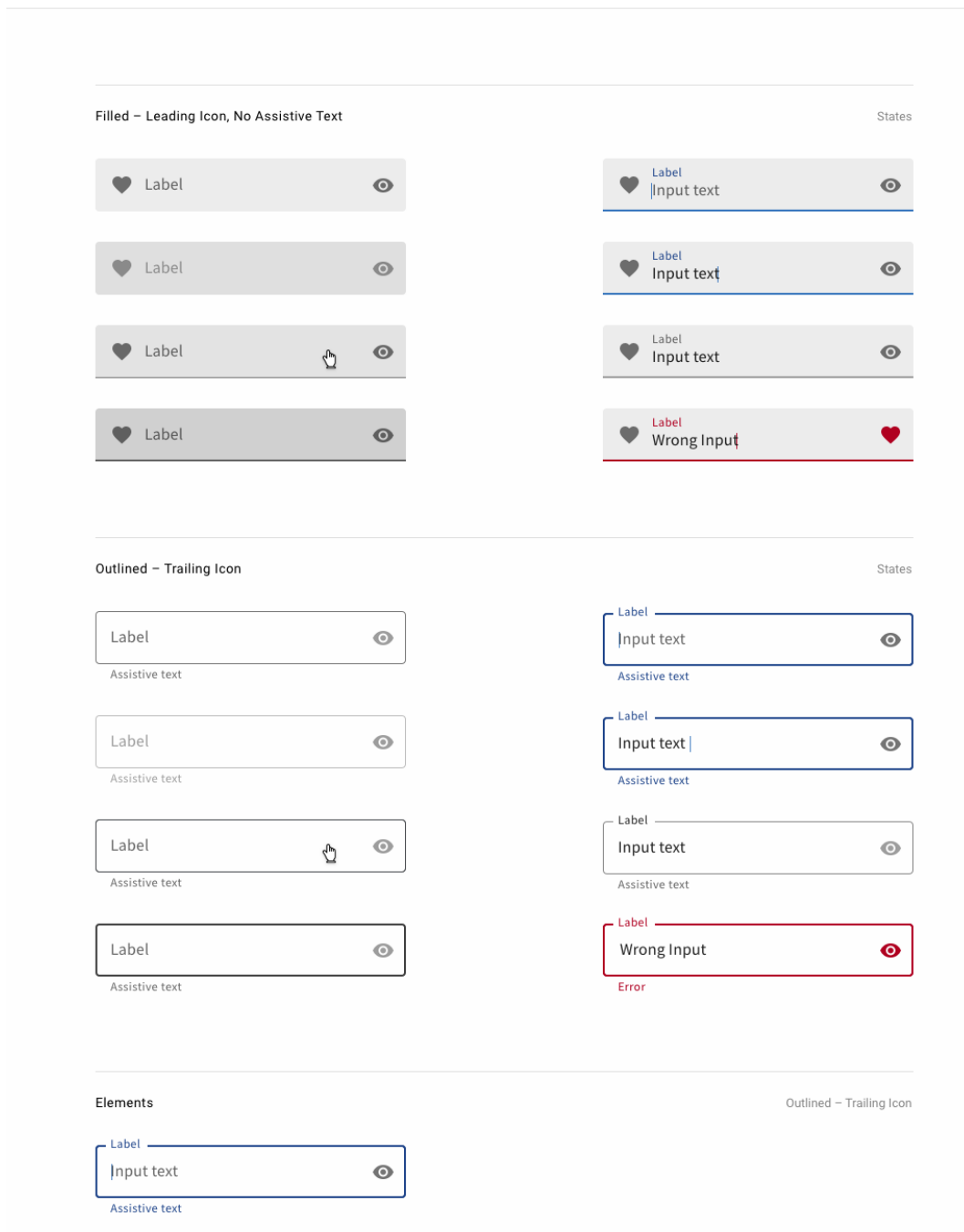


Figure 76. Text Fields (cont'd)

5 Front-end Architecture

5.1 Architecture Overview

The ELG will consist of different back-end components and solutions, which will connect their functionality in the front-end part of the ELG platform. Industry and end users require that the front-end of any system should be easy to understand and that the users are able to navigate the platform without needing any user help documentation or similar training. Although the back-end will be quite complex and modular, the front-end should be built to transform this modularity and complexity to create a simple and integrated platform that foresees seamless transition between its different parts.

While the platform should be unified from the users' perspective, the front-end should be built in a modular way from the development and platform management perspective, in order to enable fast delivery and the ability to react to change requests early. Thus, the modular approach has been chosen for the different parts of the front-end:

- 1) Single Page Application – for the dynamic content of platform: catalogue content, administrative functions, service descriptions, service trials and other functional content.
- 2) Content Management System – for the website content – training materials, news, jobs, marketing materials, landing pages and other similar stuff.

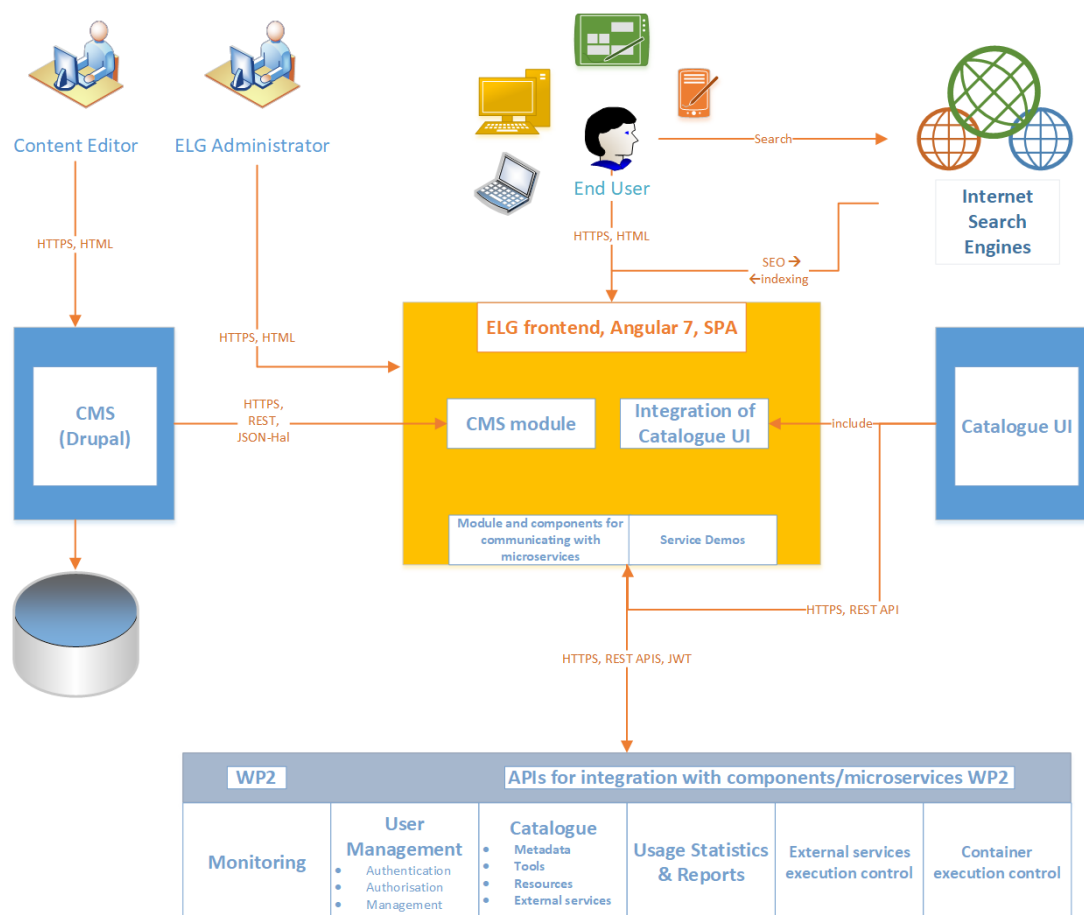


Figure 77. The Front-end architecture of European Language Grid

As end users come mostly from external internet search providers like Google, Bing, Yandex and Baidu (where they searched for a solution to their problem), the ELG platform must provide the best support for search engine crawlers.

5.2 Front-end Technology Stack

5.2.1 Choosing the Right Framework

Front-end technologies were selected based on the demanding requirements of current web users and with a best guess for longevity in the future. Our goal was to choose a set of tools for the best chance that the chosen technology will continue to be relevant and kept up to date during the project development process and in the years following its deployment in production.

Choosing the right tool stack is important for the success of the product. The requirements of a modern web platform demand cross browser compatibility, a unified look and feel on different devices including mobile, to enable an up-to-date user experience, agile development, collaboration with a graphical designer, user experience and ease of maintenance.

According to GitHub programming surveys, JavaScript leads in popularity amongst programming languages and for a couple of years has been the main choice for web projects. For a project with the scale of ELG, to select the right JavaScript framework, the following were considered – React⁹, Vue¹⁰ and Angular¹¹.

Although React and Vue are experiencing stable user growth, we settled on Angular because it has been in the market the longest (since 2010) and is the most popular choice among big and complex structured projects such as ELG will be. An important point for leaning towards Angular was that out of the three choices only Angular is considered a full framework. There is trust in this platform as it is used and developed by Google, and Microsoft is investing in different successful tools to support developer adaptation of Angular.

There have been several successful Angular-based projects in Tilde, and there is a deep knowledge base and experience in the company that confirms this choice as currently being the best of the market.

An Angular seed project has already been prepared for the needs of the ELG project and integrates tools and framework components that enrich its standard feature list and build a stable base for the front-end part of the project.

5.2.1.1 Typescript

One of the core strengths of the Angular framework is Typescript¹² – its main development language that is strongly typed. Because it is a superset of JavaScript, it is powerful but at the same time easy to adopt for JavaScript developers. Some advantages of Typescript are that it is open source, it simplifies JavaScript code (making it easier to read and debug), it can help avoid painful bugs that developers commonly run into when writing JavaScript by type checking the code, it supports static typing. All this will make development and debugging easier and save development time.

5.2.1.2 Graphical user interface theming

⁹ <https://reactjs.org/>

¹⁰ <https://vuejs.org/>

¹¹ <https://angular.io/>

¹² <https://www.typescriptlang.org/>

For graphical interface theming, we considered Twitter bootstrap¹³ and Angular Material¹⁴ by Google. Angular Material was chosen because it was created by Google to deliver modern tools to create the best user experience across websites. Bootstrap was mainly developed for dealing with responsiveness challenges across different devices. It has a very good utility library, so this is the only part that is going to be included in project. For responsive layout purposes, the flex-layout library developed by Angular is used.



Figure 78. The core of the chosen technology stack

5.2.2 Single Page Application

Our goal is to bring the best experience to the user. Lately, Single Page Applications (SPA) have advanced significantly and have advantages over Multi Page Applications. The most important aspect of SPA is that rather than re-loading each page in its entirety, an SPA application loads content dynamically. This means that page loading speed improves significantly as applications don't update the entire page but only the required content.

Angular is a framework that is meant for building SPAs, and there are several tools that help the development process and running the live application.

A single-page app can cache any local data effectively. An SPA sends only one request to a server and then stores all the data it receives. Then it can use this data and work even offline. ELG uses the ngx-cache utility as its in-app cache management tool.

5.2.2.1 Meeting SEO requirements

For a long time, SPAs provided poor SEO optimisation. Single-page apps operate on JavaScript and download data on request from the client side without changing the URL; different pages do not have a unique URL. It has been hard to optimise these websites for search engines since most pages could not be analysed by crawlers. Recently the situation has changed. There are tools that allow setting SEO data for Angular, creating SEO-friendly URLs and sharing in social networks. The tool we have chosen is ngx-meta¹⁵. It generates title, meta tags and Open Graph tags for social sharing.

5.2.2.2 Emphasis on performance

SPAs are built for performance – all HTML is sent upon the first user request, and only data is requested from the server after that. After the initial files are received, the application is built and run in the user's browser. Via

¹³ <https://getbootstrap.com/>

¹⁴ <https://material.angular.io/>

¹⁵ <https://github.com/fulls123/ngx-meta>

the mechanism of Lazy loading, Angular determines and initiates only the currently needed components so that the app can be ready as fast as possible.

5.2.2.3 Localisation support

There is a requirement for localisation support for every modern web page. Angular implements this with the help of the ngx-translate¹⁶ module. A JSON file that holds key-value pairs of translatable content can be easily translated into any language as needed.

5.2.2.4 Web application state management

In cases when an application becomes complex, because of many simultaneous data requests and its components' state changes, there is a need for a single place where to look if you want to have a clear idea of what is going on with the application. Reactive programming and state management concepts help to resolve this complexity. Reactive programming is the way to communicate between different parts of an application. Instead of pushing data directly to the component or service that needs it, in reactive programming, it is the component or service that reacts to data changes. State management is a way to write web applications so that they behave consistently and are predictable. Put simply – these two concepts help create complex web pages by creating one common application state that can be changed only through a certain, developer set protocol.

We can think of the application state as all user interface element conditions at a given moment. For example, before the user presses a button, the application is in one state and after – in another. By coordinating these interface changes centrally, we can follow user actions easily. NgRx¹⁷ is one of the libraries used for application state management. It uses store to reflect application state which can be seen as client-side database. The state is immutable, meaning that it cannot be changed directly but only through defined actions and reducers that change the state. The Reducer function takes action and previous application state and returns new state. In this way, state management can be used to log and replay user actions if needed.

5.2.2.5 Secure data exchange between client and browser

As mentioned earlier, the Angular web application will load all HTML in the browser, and data will only be exchanged between the front-end and back-end via an API after that. This exchange should be secure. For this reason, we are going to use the JSON Web Token (JWT)¹⁸ as it is an open industry standard method for transferring data. Data to be sent is encoded as a JSON object that is encrypted before being sent.

This method is good for two-way data exchange and authorisation as well. There is an Angular library that implements authentication methods using the JSON Web Token. At this stage of the project, there is no final decision on how the authentication will be managed, so a particular Angular library has not been selected yet. But if Auth0 will be used, then ngx-auth¹⁹ will be the choice.

¹⁶ <https://github.com/ngx-translate/core>

¹⁷ <https://ngrx.io/>

¹⁸ <https://jwt.io/>

¹⁹ <https://github.com/fulls123/ngx-auth>

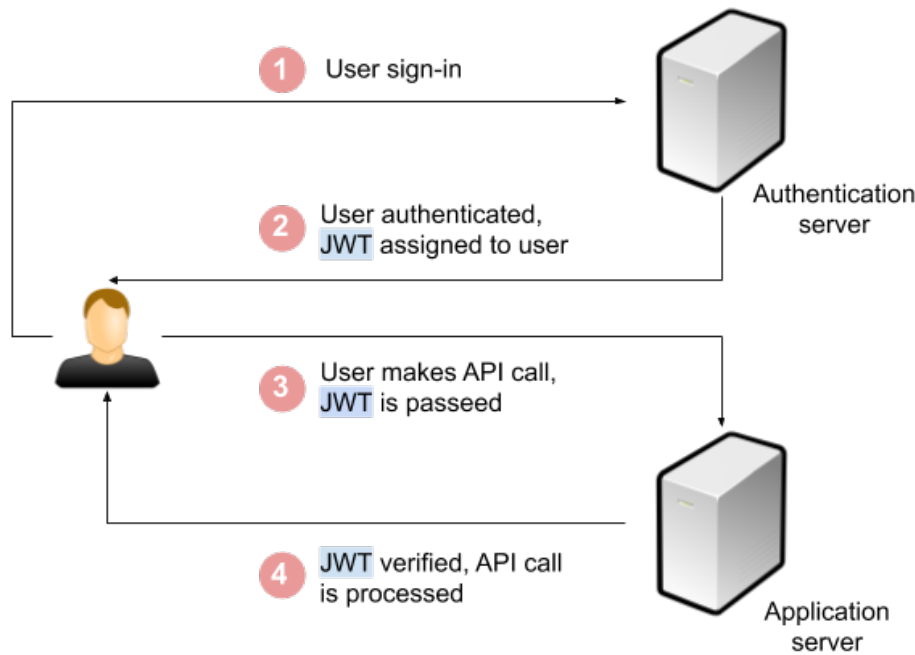


Figure 79. JWT creation and use in data exchange

5.2.3 Theming Graphical Components

For a project with the scope of ELG, it is important to maintain design and usability consistency. All user interface elements should have the same look and feel, and user experience should be flawless. Designers and usability experts have compiled all the best practices and combined them into a unified design system. The design system is essential to building better and faster. Almost every self-respecting company with a web presence has understood the need for a common design framework and has created different tools to tackle this issue – element examples, pattern libraries, style guides and so on.

Front-end developers, similarly to designers, have created various instruments to implement designs consistently and to keep up with interface updates during the lifecycle of a web page. For some time, designers and developers spoke quite different languages, and it was a challenge to find common meeting points. Material design is a great step towards unified design development and implementation.

Angular material components are implemented as an adjustable theme that can be tuned according to the designer's wishes. For example, a theme has primary and secondary colours that, once set, will be used through all interface elements. In this way, the front-end developer needs to set the theme's primary colour in just one place.

5.2.4 Tools for Development

This section lists some of the key tools that will be used in the development process.

5.2.4.1 Typescript

Since its creation in 1993, JavaScript has gone through several important updates until it has become the de facto language for web development. Nevertheless, it still is in development, and there is a variety of tools to

support it. One of these tools is the Typescript language that is a superset of JavaScript and improves the language by adding static types, interfaces, structure, prototyping and node package management.

5.2.4.2 TSLint

TSLint²⁰ is a typescript code quality tool. It checks code for readability, maintainability and functionality errors. For the developer community and for developers that work across one project, that means it is easier to understand other developers' code and to maintain code quality standards.

5.2.4.3 SCSS

SCSS²¹ gives everyday CSS scripting abilities. The developer can write functions and do other programming language-only things that make CSS code reusable and flexible. This is the main reason why theming is possible. SCSS is an extension of CSS, and the code should be translated back to CSS.

5.2.4.4 Autoprefixer

Autoprefixer²² is a plugin that parses CSS and adds vendor prefixes. This is important if the web project runs on more than one browser. While some CSS features are not yet fully supported in all browsers, vendors usually add browser-specific prefixes that work in the targeted browser but are ignored in other browsers. This plugin analyses CSS and adds the featured prefixes to it so that the outcome looks the same on every browser, where possible.

5.2.5 Tools for Building and Packing

After the code is written, it should be compiled, tested, packed for deployment and deployed.

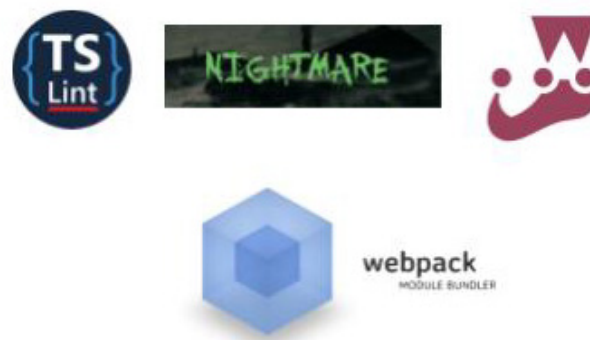


Figure 80. Testing and bundling tools

5.2.5.1 Webpack and code bundling

The current project configuration runs in Server-Side Rendering mode (SSR). The same application that is run in the client browser is pre-rendered on the server. In this way, the server sends static html code to the browser, and the user gets to see the content faster. When a page is rendered on a client's browser, it seamlessly replaces pre-rendered static html with a fully interactive Angular page. For this reason, two application bundles should be generated – one to be run on the client's browser and one to be run on the server. Webpack is the

²⁰ <https://github.com/palantir/tslint>

²¹ <https://sass-lang.com/>

²² <https://github.com/postcss/autoprefixer>

tool that bundles all the code together and generates output. It reads all the requirements, gathers dependencies and does the transformation and packing job if needed.

5.2.5.2 Tree-shaking

In environments with many libraries and code dependencies, it is common that not all code that is imported for a project is used. This is fine because code libraries speed up the development process, although they mildly slow down performance and increase final bundle size. This can be avoided by so called tree-shaking – unused code is shaken off like raindrops from leaves after the rain. This is done by webpack while packing code and creating the final bundle.

5.2.5.3 Unit testing with Jest

There are several ways to test written code quality. Jest²³ is a library for unit tests – a unit, for example an Angular component, of written code can be verified for how well it reacts to different inputs. Jest tests are written very similarly as development code and can be written once and configured to be run whenever needed, for example, before each web project deployment.

5.2.5.4 Browser testing with Nightmare

Not everything can be tested with unit tests. Unit tests are usually scoped and verify a single component. Sometimes we need to simulate user behaviour and that can be done with automated browser testing. Nightmare²⁴ is a high-level browser automation library. It uses a few simple methods that mimic user actions. Similarly to unit tests, browser tests can be written once and executed when needed.

5.3 Content Management System

To enable non-technical staff to append new information to the ELG platform, a content management system is needed. To select the most appropriate solution, the two best candidates for this purpose were initially selected based on the opinions of the open source community and previous experience. We compared two solutions – Drupal²⁵ and WordPress²⁶.

5.3.1 CMS Selection

Drupal is a modular, free and open source content management framework that is built using the PHP programming language and is available under a GNU GPL licence. The Drupal community consists of more than a million members. The standard Drupal setup consists of a set of well tested Core modules that provide basic functionality for most websites; however, the community also provides more than 40 000 free components that can extend the core functionality of Drupal.

WordPress is very simple and the most popular CMS for creating websites and blogs. WordPress is an open-sourced CMS and is available under a GPL v2 licence.

After comparing the functionality and the features available for each platform and validating them against our need for the platform's architecture, Drupal version 8 was selected based on the following factors:

- Built-in taxonomy system, supports more difficult content
- Allows scalability, working with a large amount of information

²³ <https://jestjs.io/>

²⁴ <http://www.nightmarejs.org/>

²⁵ <https://www.drupal.org/>

²⁶ <https://wordpress.com/>

- Core support for multilingual sites
- Enterprise level security
- Better with various templates for various audiences
- Better performance
- Advanced user permission system

5.3.2 Drupal Architecture

The role of the CMS is to allow the content of the European Language Grid website to be edited with the most powerful and up-to-date content authoring features:

- version control
- staged user access control
- easy content markup and design

The following CMS functionality is required for the European Language Grid:

- Uploading and publishing files, e.g., PDFs, images, documents for public download
- Management of different menu items:
 - Main menu
 - Footer menu
 - Other menus as appear in design sketches
- Information for SEO needs:
 - Metatags
 - Title, description
 - Pretty URL addresses

As the front-end will be a monolithic Single Page Application, this Content Management System will be without a dedicated public front-end. Instead of that, Drupal will serve different menus and page contents using REST services and the JSON-HAL format for data exchange. The built-in Drupal front-end will be protected by a username/password, and after successful authentication, the user will be redirected to the content management environment and authorised for specific operations based on an assigned role.

5.3.3 Selected Drupal Components

The following Drupal 8 core modules will be used to set-up the European Language Grid CMS:

- Automated Cron – Provides an automated way to run cron jobs by executing them at the end of a server response.
- Block – Controls the visual building blocks a page is constructed from. Blocks are boxes of content rendered into an area or region of a web page.
- Breakpoint – Manages breakpoints and breakpoint groups for responsive designs.
- CKEditor – WYSIWYG editing for rich text fields using CKEditor.
- Configuration Manager – Lets administrators manage configuration changes.
- Contextual links – Provides contextual links to perform actions related to elements on a page.
- Custom Menu Links – Allows administrators to create custom menu links.
- Database Logging – Logs and records system events to the database.
- Field – Field API to add fields to entities like nodes and users.
- Field UI – User interface for the Field API.

- Filter – Filters content in preparation for display.
- Help – Manages the display of online help.
- History – Records which user has read which content.
- Internal Dynamic Page Cache – Caches pages for any user, handling dynamic content correctly.
- Internal Page Cache – Caches pages for anonymous users. Use when an external page cache is not available.
- Menu UI – Allows administrators to customise the site navigation menu.
- Node – Allows content to be submitted to the site and displayed on pages.
- Path – Allows users to rename URLs.
- RDF – Enriches your content with metadata to let other applications (e.g., search engines, aggregators) better understand its relationships and attributes.
- Shortcut – Allows users to manage customisable lists of shortcut links.
- System – Handles general site configuration for administrators.
- Taxonomy – Enables the categorisation of content.
- Text Editor – Provides a means to associate text formats with text editor libraries such as WYSIWYGs or toolbars.
- Toolbar – Provides a toolbar that shows the top-level administration menu items and links from other modules.
- Update Manager – Checks for available updates and can securely install or update modules and themes via a web interface.
- User – Manages the user registration and login system.
- Views – Creates customised lists and queries from your database.
- Views UI – Administrative interface for Views.

For the system to best fit our requirements, additional community contributed modules were installed and configured:

- HAL – Serialises entities using Hypertext Application Language.
- REST menu items – This module provides a REST endpoint to retrieve menu items based on the menu name.
- REST UI – Provides a user interface to manage REST resources.
- RESTful Web Services – Exposes entities and other resources as RESTful web API.
- Serialisation – Provides a service for (de)serialising data to/from formats such as JSON and XML.
- Libraries – Allows version-dependent and shared usage of external libraries.
- Advanced Link – Adds title, target, etc. attributes to Text Editor's link dialog if the text format allows them.
- PathAuto – Provides a mechanism for modules to automatically generate aliases for the content they manage.
- Token – Provides a user interface for the Token API and some missing core tokens.
- Imce File Manager – Provides a file manager supporting personal folders.
- Relative to Absolute Filter – Filters for conversion of relative paths to absolute URLs.
- Chaos tools – Provides a number of utility and helper APIs for Drupal developers and site builders.
- CKEditor CodeMirror – Adds the CKEditor CodeMirror source syntax highlighter plugin.

In case the system will require support for multilingual content, the following modules have been prepared to be enabled for this case:

- Configuration Translation – Provides a translation interface for configuration.
- Content Translation – Allows users to translate content entities.
- Interface Translation – Translates the built-in user interface.
- Language – Allows users to configure languages and apply them to content.

5.3.4 Menu Generation and Serving

Drupal will have various menus and different link sets using the REST API interface. The complete set of menus will be finished during the interactive development process. Initially, there are two different menus available – the main menu and the footer menu. The footer menu is a hierarchical list of links, which resides in the footer of the European Language Grid portal.

```
▼ [,...]
  ▶ 0: {key: "95c5925a-53f3-40be-99ff-32840328ccb3", title: "CMS page", description: null, uri: "node/84",...}
  ▶ 1: {key: "f7dd513f-316d-4ffa-b72d-4c2fa29c1f49", title: "Angular page", description: null,...}
  ▼ 2: {key: "2be69e03-42d1-4138-b4e4-e1602ef16ed6", title: "External page", description: null,...}
    absolute: "https://example.com"
    alias: null
    description: null
    enabled: true
    existing: true
    expanded: false
    external: true
    key: "2be69e03-42d1-4138-b4e4-e1602ef16ed6"
    options: {external: true}
      external: true
      relative: "https://example.com"
      title: "External page"
      uri: "https://example.com"
      uuid: null
      weight: "0"
  ▼ 3: {key: "f88b37a2-f4c8-4ce5-b706-de114864be40", title: "UI component showdown", description: null,...}
    absolute: "http://trupal/ui-showdown"
    alias: null
    description: null
    enabled: true
    existing: true
    expanded: false
    external: false
    key: "f88b37a2-f4c8-4ce5-b706-de114864be40"
    options: []
      relative: "/ui-showdown"
      title: "UI component showdown"
      uri: "base:ui-showdown"
      uuid: null
      weight: "0"
```

Figure 81. Example of the Drupal menu structure, JSON-HAL format

5.3.5 Content Serving

There will be authored content that is stored in the Drupal database (HTML fragments, metadata) and shared public files that are uploaded to the fileshare of the Content Management System.

```
{
  "_links": {
    "self": {
      "href": "http://trupal/sakums?_format=hal_json",
      "type": "application/hal+json"
    },
    "nid": {
      "value": 3
    }
  },
  "changed": {
    "value": 1534428314,
    "lang": "en"
  },
  "created": {
    "value": 1504679221,
    "lang": "en"
  },
  "default_langcode": {
    "value": true,
    "lang": "en"
  },
  "field_sekcija": {
    "value": {
      "format": "full_html",
      "processed": "<div class='container'><div class='row'><div class='col-md-12'><h2>Aicinām iesaistīties arī tevi</h2></div></div><div class='row'><div class='container'><div class='row'><div class='col-md-12'><h2>Aicinām iesaistīties arī tevi</h2></div></div></div></div><div class='row no"
    }
  },
  "langcode": {
    "value": "en",
    "lang": "en"
  },
  "nid": {
    "value": 3
  },
  "path": {
    "alias": "/sakums",
    "pid": 3,
    "langcode": "en",
    "lang": "en"
  },
  "promote": {
    "value": false,
    "lang": "en"
  },
  "revision_timestamp": {
    "value": 1534428314
  },
  "revision_translation_affected": {
    "value": true,
    "lang": "en"
  },
  "status": {
    "value": true,
    "lang": "en"
  },
  "sticky": {
    "value": false,
    "lang": "en"
  },
  "title": {
    "value": "Sākums",
    "lang": "en"
  },
  "type": {
    "target_id": "sekciju_lapa"
  },
  "uuid": {
    "value": "02f20ea5-c87f-4f5a-9931-7e1307937a85"
  },
  "vid": {
    "value": 237
  },
  "_embedded": {
    "self": {
      "href": "http://trupal/sakums?_format=hal_json",
      "type": "application/hal+json"
    }
  }
}
```

Figure 82. Example of returned content page by CMS

Authored content will be served using the REST API and JSON-HAL formats. The Angular SPA front-end application will transform the data into a user scenario specific layout and portal design. Static files like PDFs, photos, graphics and images will be provided and accessed directly from the CMS system.

5.4 API Requirements

As a Single Page Application, the described front-end solution requires all backend services to be accessible through the internet directly or any other public service gate/proxy. When planning the endpoint of service, the following must be taken in account:

- Authentication and Authorisation. Token based Authorisation, such as JSON Web Tokens (JWT), is suggested.
- Accessible through the internet. All services must serve data needed for the User Interface and other data for front-end.
- 24/7 availability – all microservices should be operational all the time; malfunctioning of one microservice due to its erroneous state or undergoing maintenance can lead to malfunction of all the front-end UI.
- Stateless – all services should be stateless (should treat each request as an independent transaction that is unrelated to any previous request), unless it is designed in an asynchronous manner. In the case that a service is taking too much time to complete, an asynchronous approach could be created.
- RESTful (Representational State Transfer) services are preferred. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations²⁷.

²⁷ https://en.wikipedia.org/wiki/Representational_state_transfer

- Services must be compliant with a selected user scenario and graphical and functional design. If there is a specific group of data or functions to illustrate in certain scenario step, then a dedicated micro-service should return at least these data and provide appropriate operations for the scenario.

5.5 Deployment of the Front-end

All components of the front-end solution will be containerised and delivered to be ready for deployment in the container orchestration environment – Kubernetes²⁸. In addition, a Kubernetes deployment configuration using Helm²⁹ will be created during this action.

Azure DevOps³⁰ was chosen for front-end development, issue tracking, code repository and the build pipeline. The developed source code is checked into the code repository, then it is checked out, and the Docker image build process is started. After the image is built, it is tagged with a qualified tag and pushed to the private Azure Container registry.

All the built Docker images will be used in the infrastructure provided by WP1 and contain the Helm chart, which is dedicated to the front-end solution deployment and setup.

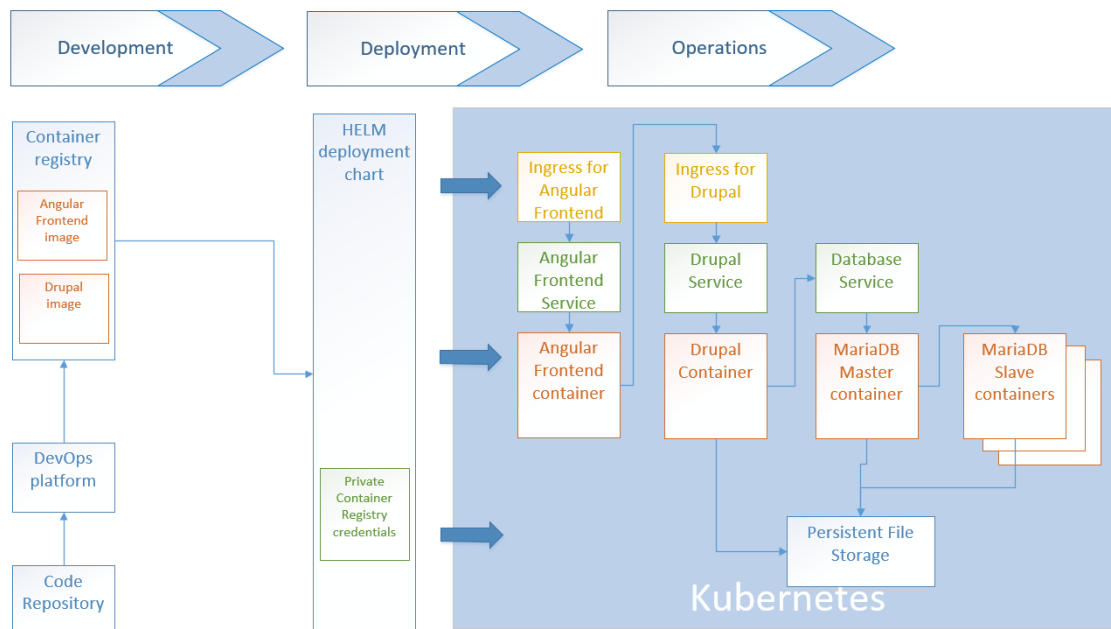


Figure 83. Code delivery and deployment strategy of ELG front-end

Deployment of the Dockerised solution will contain:

- Deployment
- Services
- Ingress
- Chart for Database
- Persistence volume claims

²⁸ <https://kubernetes.io/>

²⁹ <https://helm.sh/>

³⁰ <https://azure.microsoft.com/en-us/services/devops/>

Annexes

Annex I. User Survey

In January 2019, the EU-funded project European Language Grid (ELG) was launched. The aim of this project is to develop the first – and primary – platform for European Language Technologies, and we want it to become the most important digital marketplace for language technology solutions in Europe.

Not only will it be a business directory of companies, universities, research centres and independent experts. The platform will also contain a repository of thousands of data sets as well as hundreds of tools and services that can be used directly from the grid. In order to make this platform as useful and as efficient as possible for its users – that is, for you – we ask you to provide your valuable feedback in this survey. It will take approximately 15 minutes to complete, and you can be sure that your data are treated strictly confidentially. Thank you very much for your support!

Participant Profiling

1. What is your organization type?
 - a. Academic/ Research
 - b. Industry, Large
 - c. Industry, SME
 - d. Start-up
 - e. Public Administration
 - f. Non-Governmental Organisation
 - g. Self-employed
 - h. Other (please specify)
2. What is your primary role in the organization?
 - a. Developer
 - b. Researcher
 - c. Business manager
 - d. Research/academic manager
 - e. Business owner
 - f. Sales manager
 - g. Marketing manager
 - h. Public officer
 - i. Other (please specify)
3. In which of the following activities is your company/organization involved (select all that apply)?
 - a. Language technology development
 - b. Language technology research
 - c. Language technology services
 - d. Academic teaching in the natural language processing, computational linguistics or related area
 - e. eCommerce
 - f. Localisation
 - g. Data analytics
 - h. Other (please specify)

4. Have you ever provided your language data (e.g., corpora, lexica, terminologies, etc.) or software/tools to any online platform (GitHub, META-SHARE, CLARIN, ELRA catalogue, etc.)
- a. Yes
 - b. No
 - c. Don't know

Content submission branch

5. Have you ever submitted **language data (datasets, corpora, etc.)** through specific repositories/registries? Which ones? (select all that apply)
- a. Institutional repository, Institutional/personal website
 - b. CLARIN repository
 - c. ELRA catalogue
 - d. META-SHARE
 - e. DataHub
 - f. figshare
 - g. zenodo
 - h. Software repository (GitHub, gitlab, etc.)
 - i. I haven't submitted language data before
 - j. Other (please specify)
6. Have you ever submitted **language processing software tools/services** through specific repositories/registries? Which ones? (select all that apply)
- a. Institutional repository, Institutional/personal website
 - b. CLARIN repository
 - c. META-SHARE
 - d. DataHub
 - e. figshare
 - f. zenodo
 - g. Software repository (GitHub, gitlab, docker hub, etc.)
 - h. I haven't submitted language tools/services before
 - i. Other (please specify)
7. What types of content (data or software/tools) have you submitted? (select all that apply)
- a. Tools/ Services (any type of software)
 - b. Textual datasets/ Collections of text data/ Corpora
 - c. Audio datasets/ Collections/ Corpora
 - d. Video datasets/ Collections/ Corpora
 - e. Annotated datasets/ Collections/ Corpora
 - f. Lexica, Terminologies, etc.
 - g. Ontologies, Vocabularies, Glossaries, etc.
 - h. Models, Computational grammars, etc.
 - i. Sign language datasets
 - j. Other (please specify)
8. What was your motivation when submitting content (select up to 3)
- a. Promotion of the content

- b. Project requirements
 - c. Promotion of the organization
 - d. Financial interest
 - e. Goodwill (to serve community, etc.)
 - f. To ensure reproducibility of the research results
 - g. Other (please specify)
9. How is your content (data/ software tools or services) documented when shared? Do you use any specific metadata standards or schemas? If yes, which ones?
- a. Dublin Core
 - b. OLAC
 - c. DCAT
 - d. CKAN
 - e. schema.org
 - f. META-SHARE
 - g. CMDI profile
 - h. Metadata standards/schema were not necessary
 - i. Other (please specify and provide a link or a reference to the metadata schema)
10. Have you experienced any difficulties in sharing your content? (select all that apply)
- a. Everything was fine, no specific difficulties experienced
 - b. Do not remember or was not personally involved
 - c. Process was too time-consuming
 - d. It was not clear what to do or how to fill the required information
 - e. Too many fields to fill
 - f. Lack of some metadata categories that are specific to my dataset/software
 - g. No possibility to insert/modify metadata by myself Unclear which license to assign
 - h. No support
 - i. Platform was not performing as expected or was too slow
 - j. Did not know which data center to work with
 - k. Copyright/ IPR issues prevented from sharing
 - l. Other (please specify)
11. What do you like about the platform in which you shared your content? (select all that apply)
- a. Positive experience in general
 - b. Intuitive/Self-explanatory process
 - c. Not too many obligatory fields to fill
 - d. Rich metadata – all the categories available to provide full metadata information about my datasets/software
 - e. Clear guidance to select the most suitable type of licence
 - f. Possibility to add documentation
 - g. Possibility to show examples or demo/trial
 - h. Option for user to leave review/ feedback
 - i. User ratings
 - j. Usage statistics (e.g., number of downloads, views)
 - k. Platform is well-known/well-recognized

- l. No requirement for exclusiveness
 - m. Possibility to insert/modify description by myself
 - n. Human support provided
 - o. Other (please specify)
- 12. What parameters are the most important for you to measure the success of sharing your content? (select up to 3 most important)
 - a. Number of downloads
 - b. Number of views
 - c. User ratings/reviews
 - d. References in publications
 - e. Financial income from selling the data/software
 - f. Redirects to company website
 - g. Number of contact from potential users
 - h. None
 - i. Other (please specify)
- 13. How important for you are the following factors when submitting your content (evaluate each factor on the scale from 1 – Not important at all to 4 – Very important)
 - a. The process is easy and quick
 - b. Possibility to provide all metadata that relate to the particular data/software
 - c. Availability of a wizard allowing to define the license that best fits my intents
 - d. Support is available
 - e. Popularity of the platform
 - f. Usage statistics about the number of views or downloads
 - g. Preserving and providing access to older versions of data/software
 - h. Platform interface on all EU languages
 - i. Secured access

Containerized software

European Language Grid will employ a novel approach of containerization. It goes as follows: the provider creates a Docker image (which contains the code and all its system libraries, system tools, dependencies etc.) and uploads it to the image repository. This Docker image can be run on any system through a virtual machine. When the user wants to run the software, the image is deployed on the ELG cloud or any other third party cloud (or downloaded and deployed on the user's system). Users of the software do not need to worry about any issues of interoperability.

- 14. Are you interested in the option to provide your software as a container?
 - a. Yes
 - b. No
 - c. Don't know
- 15. If you would like to provide a containerized software, what kind of support would you need? (select all that apply)
 - a. Written instruction is enough
 - b. Video tutorial
 - c. Email support

- d. Live support
 - e. "I provide the code – you make container out of it"
 - f. I don't plan to use a containerized software
 - g. I don't plan to use support
 - h. Other (please specify)
16. Would you be ready to pay for support that gives advice on how to best integrate your software in ELG?
- a. Yes
 - b. No
 - c. Other (please specify)
17. Would you be ready to pay for support that can create a container out of your code?
- a. Yes
 - b. No
 - c. I don't plan to use containers
 - d. Other (please specify)
18. If you plan to provide a containerized tool/software, would you prefer to:
- a. Deploy your software through the ELG platform
 - b. Link to a remote repository where your software is hosted
 - c. I don't provide tools/software
 - d. Other (please specify)

Paid Content Submission

19. Would you want to provide a paid content (to sell) through ELG?
- a. Yes
 - b. No
 - c. Don't know
20. How would you prefer to handle the payment owed to you (select all that apply).
- a. Yourself
 - b. Through ELG as an intermediary
 - c. I don't know
 - d. Other (please specify)
21. Would you be ready to pay a fee to a platform that sells your software?
- a. No
 - b. Yes, a commission fee up to 10%
 - c. Yes, a commission fee 10-20%
 - d. Yes, a commission fee 20-30%
 - e. Yes, a commission fee 30-40%
 - f. Yes, a commission fee of more than 40%
 - g. Fixed amount per month
 - h. Other (please specify)

Participant Profiling

22. Have you ever searched language data or software/tools on such platforms as META-SHARE, ELRC-SHARE, CLARIN, ELRA etc.?

- a. Yes
- b. No
- c. Don't know

Content Usage

23. What types of content (data/software) have you searched for? (select all that apply)

- a. Tools/ Services (any type of software)
- b. Textual datasets/ Collections of text data/ Corpora
- c. Audio datasets/ Collections/ Corpora
- d. Video datasets/ Collections/ Corpora
- e. Annotated datasets/ Collections/ Corpora
- f. Lexica, Terminologies, etc.
- g. Ontologies, Vocabularies, Glossaries, etc.
- h. Other (please specify)

24. Where have you searched for **language data (datasets, texts, etc.)**? Please select all that apply

- a. Institutional repository
- b. CLARIN repository
- c. DataHub
- d. ELRA catalogue
- e. META-SHARE
- f. Figshare
- g. Zenodo
- h. Software repository (GitHub/ gitlab etc.)
- i. General search engine
- j. Google dataset
- k. LDC (Linguistic Data Consortium)
- l. I haven't searched for language data before
- m. Other (please specify)

25. Where have you searched for **language processing software tools/services**? Please select all that apply

- a. Institutional repository
- b. CLARIN repository
- c. DataHub
- d. META-SHARE
- e. Figshare
- f. Zenodo
- g. Software repository (GitHub/ gitlab/ docker hub etc.)
- h. General search engine
- i. Google dataset
- j. I haven't searched for language tools/services before
- k. Other (please specify)

26. Please choose the most important criteria you use when searching for **language data** (e.g., datasets, corpora, lexica, terminologies, etc). Select up to 5
- Name
 - Price
 - Language of the contents
 - Available documentation
 - License/access conditions
 - Domain/topic/type of data (e.g., finance, politics, biology, tweets, etc.)
 - Size of dataset
 - Data type/format (e.g., RDF, Wave, MPEG, Text/Utf-8, etc)
 - Ability to preview
 - Other (please specify)
27. Please choose the most important criteria you use when searching for **language processing software tools/services** (select up to 5)
- Name
 - Price
 - Performance
 - Reviews
 - Language coverage
 - Programming language
 - Usability
 - Available documentation
 - License/access conditions
 - Availability of open source code
 - Described in/attached to a scientific publication
 - Ability to test the tool/service
 - Other (please specify)
28. If you use free text queries for searching, could you provide few examples of such queries? (optional question)
29. What would you prefer for using the **software service/tool**? (select all that apply)
- Download executables or code
 - Download containerized tool/service
 - Via vendor-specific API
 - Via standardized API aggregating services from multiple vendors
 - Use service through web user interface
 - Other (please specify)
30. How important it is for you to have the older versions of the same content (data/software) available on the ELG? (estimate on the scale from 1 – Not important at all to 4 – Very important)
31. Have you experienced any difficulties in searching for language content on existing platforms (META-SHARE, ELRA, CLARIN, GitHub, etc.)? Please select all that apply.
- Everything was fine, no specific difficulties experienced
 - Do not remember or was not personally involved
 - Could not find necessary data/software

- d. Too many irrelevant results
- e. Too complex description and/or metadata information
- f. Insufficient description and/or metadata information
- g. Unclear licence and usage conditions
- h. No support
- i. Platform not performing as expected or was too slow
- j. Data/software did not correspond to its description
- k. Lack of samples/ demo/ trial
- l. No option to leave review/ feedback
- m. Other (please specify)

Other Questions

32. Would you be willing to **contribute** the following types of language content

- a. Corpus (sets of texts, e.g., articles, legal acts, parallel corpora, translation memories; sets of audio recordings, videos in one or more languages, etc.)
- b. Language description (computational grammars, language and translation models, machine learning models, etc.)
- c. Lexical/conceptual resource (terminological lexica, glossaries, lists of words, names, places, dictionaries containing words with linguistical information, etc.)
- d. Tool or Service (language processing tools, web services, applications such as linguistic annotators, e.g., part-of-speech taggers, parsers, etc., semantic annotators, e.g., named entity recognisers, aligners, etc.)
- e. None
- f. I don't know
- g. Other (please specify)

33. Would you be interested to **use** the following types of language content

- a. Corpus (sets of texts, e.g., articles, legal acts, parallel corpora, translation memories; sets of audio recordings, videos in one or more languages, etc.)
- b. Language description (computational grammars, language and translation models, machine learning models, etc.)
- c. Lexical/conceptual resource (terminological lexica, glossaries, lists of words, names, places, dictionaries containing words with linguistical information, etc.)
- d. Tool or Service (language processing tools, web services, applications such as linguistic annotators, e.g., part-of-speech taggers, parsers, etc., semantic annotators, e.g., named entity recognisers, aligners, etc.)
- e. None
- f. I don't know
- g. Other (please specify)

34. How important would it be for you to have one centralized digital meeting spot for LT in Europe? (estimate on the scale from 1 – Not important at all to 4 – Very important)

35. Are you interested in paid content (data/software)?

- a. Yes, I'm ready to pay for good content
- b. No, I'm searching only for free content

- c. Don't know
 - d. Other (please specify)
36. What type of services would be the most useful for you (select all that apply)?
- a. Process and retrieve small portions of information and get response immediately (synchronous requests)
 - b. Upload large bunch of data and later retrieve results (asynchronous requests)
 - c. Both are needed
 - d. It is not important, any of the methods would be acceptable
 - e. I cannot answer that question
 - f. Other (please specify)
37. How useful it would be for you to have more information about (estimate each factor on the scale from 1 – Not important at all to 4 – Very important)
- a. Trainings
 - b. Events
 - c. Job advertisements
38. Would you agree to create a full profile for yourself and/or your organization and populate it with information?
- a. Yes
 - b. No
 - c. Don't know
 - d. Other (please specify)
39. Could you suggest any specific recommendations for ELG based on your experience with industry software service platforms (e.g., NPM, NuGet, AWS Marketplace, SDL AppStore or PlayStore/ Apple AppStore, etc.)?
40. In 2020, ELG will announce two open calls for demonstrator projects, where partial funding for development of innovative applications/technologies will be offered. Would you be interested in (select all that apply):
- a. taking part in the open call and submit a demonstrator project
 - b. getting involved in the evaluation of the project proposals
 - c. getting more information on the open calls for demonstrator projects
 - d. none of the above
 - e. Other (please specify)

Contact Information

41. Please provide your contact information (can be left blank)
- a. Full name
 - b. Company
 - c. Email
42. Would you like to receive information about the ELG platform
- a. Yes
 - b. No
43. Would you be interested to participate in the ELG platform development process?
- a. to participate in the testing and providing feedback on early releases of the platform

- b. to participate in an phone interview to better understand the needs of various user groups
- c. No
- d. Don't know

44. I hereby declare that I agree to the computation and use of this data for the ELG project. Any other use or processing is not permitted.

- a. Confirm

Annex II. Interview Notes

Interviewee 1.

Translation provider/founder, responsible for tooling

Interviewer's notes

Here is an example that finding the right tooling even when being in the translation industry might not be so easy. There is a huge terminology and knowledge barrier between industry and LT tool providers. Non-transparent pricing and security issues.

Needs

- Tailored and secure machine translation tool

Motivation

- Gain a competitive advantage
- Make sure she has made the right decision
- Remaining relevant in the age of AI

Questions

- Can I trust this tool/company?
- Is this secure?
- How much does it cost?
- How does machine translation work?
- What do I have to do so that I can get a good solution?
- What is up with the license?

Existing pain points

- Non-transparent pricing
- Wasting time when talking with customer reps
- No clear understanding of how Language tools work (relies on tool sellers for explanation)
- Not familiar with advanced LT lingo

Happy moments

- A killer demo

Notes

- Law firms and lawyers verify whether the translated text is ok
- Relies on customer feedback for the quality of translation
- Looked in National association for Machine translation to get trustworthy results
- When interested, connects through email
- Failed with chatbot
- Needs advice on what to choose
- Wants to be guided and be able to trust
- Clarity is priority

- Maybe a questionnaire/quiz that gives the right solutions
- Clear categories
- Would like to see corpora contents to see whether they are relevant because has had experience with suspicious segments
- Understands that machine translation is in its early stage but wants to be an early adopter and is more motivated to go through all the hassle

Interviewee 2.

LT Corpora owner / curator

Interviewer's notes

For his specific use case, it is hard to find good data online, mostly has to create it himself. Mentions that data owners might not be aware that they own something valuable, and legal/license issues might stop data owners to submit it. Has submitted data to repositories before, mentions that needs to be familiar with the dataset to fill in the meta-information.

Needs

- Good data to train machine learning algorithms

Motivation

- Gain a competitive advantage
- Submitted corpora only because of project requirements, but could not find time to update it
- Earn money from selling corpora, however might not outweigh the competitive drawbacks

Questions

- Is this data any good for me?
- Should I invest time in this data?
- Will this data produce the output I want?
- What is up with the license?
- What languages are supported?
- Is there any documentation?

Existing pain points

- It takes work to structure data correctly
- Data owners are confused with the legal side of licensing
- Has to update corpora that was submitted as a requirement of a project when errors are discovered but has no time because he's working on other projects
- It is hard to fill in all the meta-information if not familiar with the data or didn't make the corpus

Happy moments

- The submitted tool is actually used by someone else

Notes

- Mentions that some data is owned by organisations who are mostly unaware that this data might help an LT developer. Although the new EU regulation about public data availability might help with this, he says there might be a problem with the legal issues.

Interviewee 3.

LT Developer / Commercial (Big or Small)

Interviewer's notes

Interviewee is the typical LT developer, working in the industry. He mentions that he doesn't use the existing LT portals but searches on GitHub. He is interested in reliable up-to-date tools. Checks out academic resources only to find cutting edge tools. Mentions competitive advantage; therefore, he won't share his own tools.

Needs

- Tools to improve LT tools for commercial solutions

Motivation

- Gain competitive advantage

Questions

- What is up with the license?
- What languages are supported?
- What programming language is this made in?
- Can I rely on this tool? (but would sacrifice reliability for cutting edge technology)

Existing pain points

- Tools are not being updated
- Some tools require registration and are behind a paywall, limited to academia
- Some tools lack proper documentation, usually are poorly maintained

Happy moments

- Quickly finding the right tools

Notes

- Would not share tools because of competitive advantage, only if it is the requirement of the project
- Academics might publish something for reputations and references
- Wants to make the best tools in the industry to be competitive
- First looks in Google, then GitHub, if not there then META-share
- If needs cutting edge tech then searches Google Scholar or papers with code
- META-share has too many metadata, no clear categories
- Performance is important too
- The tool should have a good rating

Interviewee 4.

LT Developer, speech recognition, previously an academic researcher

Interviewer's notes

Interested in quality data to train AI. Finding a quality dataset saves a lot of time for development. Mentions lack of data for small languages and dialects. From the academic standpoint – tools are shared not only for academic reputation but also for carrying on code created during research so that it could be built upon.

Needs

- Quality corpora/dataset in the required dialect
- Storing tools created during academic research

Motivation

- Train speech recognition algorithms for clients
- Improve accuracy for speech recognition to gain competitive advantage
- Academic recognition
- Receive feedback when submitting tools
- Networking with other users of the tool
- So future students might build upon the existing findings

Questions

- What is up with the license?
- How much does this dataset cost? Is there any difference between academic and commercial use?
- What language/dialect is the corpora?
- What are the transcription guidelines? Is the dataset transcribed in the way I need it to be?
- Is this dataset somehow connected to any other datasets?
- What version is the dataset/corpora?

Existing pain points

- Corpora is bad quality or not the right dialect, wrong encoding
- Building own quality corpora takes a lot of time
- Unclear licenses, some datasets are not available for commercial use
- Tools go out of date and are not compatible with existing technologies

Happy moments

- Benchmarking LT and finding out it performs better
- Saving time by getting good corpora

Notes

- Works with the Arabic language, so dialects are very important. Some corpora sites do not list the dialects of the corpora. Mentions an example with Spanish – Mexican and European Spanish have different vocabularies.
- Needs to store existing code built during research, so the next batch of students can build upon the previously done work
- Looks for corpora in ELRA first, sometimes SpeechOcean
- Needs a good readme file that lists encoding and transcription guidelines. Some cases have missing data, so he checks samples for more details.
- There is a difference in how speech is recorded – it could be from broadcasting or phone conversations, which impacts quality.
- Building models is easier than finding the right corpora
- Recognises Creative Commons licenses and would wish to clearly see if the dataset can be licensed for both academic and commercial use
- A quality dataset saves time in building the actual solution
- Uses KALDI and participates in the open-source community but builds own add-ons that serve their competitive advantage
- Subscribes to KALDI newsletter

- Would like to see if there are any other datasets connected, either different parts or different versions of the same data
- Mentions that once found a good dataset, but had to email back and forth to find out about commercial use license, which was unclear at first
- Learns about news in the industry from conferences and newsletters, such as ELRA, KALDI, GitHub community
- When asked about the differences between academia and industry, says that academic tools are basic and straight to the point, but industrial tools have been tested for commercial applications and are more secure

Interviewee 5.

Responsible for governmental language technologies in the area of defence and government intelligence

Interviewer's notes

Not responsible for tooling and daily activities don't include looking for corpora or language tools, but provided a really great broad context of the language problems in Europe.

Needs

- Multilingual tools for understanding the context of online text
- Multilingual tools for communicating online
- Scraping online text in search of new vocabulary
- High-quality content repositories to scrape
- Information about events and news in the LT community

Motivation

- Inform the general public within the European Union and worldwide
- Understanding what is going on in online media both within Europe and outside of the EU
- Keep up to date with current LT

Existing pain points

- There are no good multilingual solutions out there
- Tools are incompatible
- Lack of high-profile data repositories in smaller languages
- Good tools are very expensive and might not meet the expectations

Happy moments

- Seeing LT doing good for society, informing and preventing possible catastrophes
- Getting good quality data to work with

Notes

- Relies on experts in deciding which language tools to use/purchase
- Internet language might differ from common language, this might impact how well online texts are understood, especially conversations and social media. Requires tools to be able to understand such texts.

- There are more than 200 language dialects around Europe, and there are no language data or language tools for some of these. This is a big struggle since online text cannot be analysed, nor is it possible to communicate in these dialects.
- Also uses statistical methods to determine impact
- At the moment, uses Deep-L, IBM solutions, needs the best there is but is limited by budget. Tools made by big companies are not compatible with each other.
- The more complex the tooling, the more compatibility becomes an issue
- Needs tools to work fast. Particularly interested in catastrophe analysis, multilingual communication, social media analysis, real-time monitoring, quality content repositories, AI tools, text analysis tools, speech-to-text and speech recognition, social topic maps. Sentiment analysis.
- Licenses are important, since the budget is limited
- Small languages have small corpora and it is not possible to build on that
- Tools are not inter-compatible. No standardisation. For example, there might be a really good tool for French, but it has poor German support. These tools come from competing companies that are not interested in a common standard; therefore, good multilingual tools don't exist. Improving the compatibility of tools with standardisation would allow having a modular tool approach that would work well for multilingual applications.

Interviewee 6.

Freelance media consultant for large advertisers

Interviewer's notes

Does prove that there is a lack of a good source of information. Relies on networking to find the best tool available because there is a lot of noise and no reliable source of information online. Has previous experience of looking for a good speech to text tool but was unable to find anything due to the bad output quality.

Needs

- A valuable source of information about current language technologies available
- Good use cases
- To find out about upcoming LT events
- Networking with experts in the LT industry

Motivation

- Remain an expert in LT, keeping up to date with the news in the industry
- Provide good advice to clients

Existing pain points

- At the moment, there are no good and reliable sources of information on LT
- The LT technology community seems like an exclusive group
- Hard to find reliable providers of quality LT tools
- Non-transparent pricing
- A lot of noise, hard to find a solution that actually works, therefore relies on networking
- Lack of good content about LT
- LT technologies can't provide quality outputs

Happy moments

- Giving valuable advice to a particular client
- Seeing LT in action

Questions

- What LT technologies currently exist?
- Which are the best tool providers out there in the current market?
- How capable are LT technologies at the moment? What is possible?
- How can LT technologies be used in practical applications?

Notes

- Relies on word-of-mouth and networking for finding the right tools
- A frequent visitor of LT conferences and events – networks a lot
- Clients are mostly large organisations interested in advertising in non-English media
- Has previous experience in looking for a speech to text tool but was unable to find a good quality solution
- Does media strategies – large companies are interested in how their values relate to the values of the media they are advertising on. For example, one media source might be politically skewed, which might become a huge trade-off for a large corporation that has politically neutral values.
- Mentions that interested in: Webinars, speech to text, truth verification, face recognition, diction, understanding LT features, use cases, would like to try out LT tools, transparent pricing