THE

INITIATIVE

Language Interfaces on the Web

http://lionweb.io
info@lionweb.io

MISSION

To create an ecosystem of interoperable components for building language-oriented modeling tools on the web.

We believe that a lively ecosystem will facilitate the adoption of language engineering and modeling solutions by reducing vendor lock-in. Development of advanced solutions will be accelerated by mixing and matching components, potentially sourced from different vendors or open source initiatives. It will also foster innovation, as incremental improvements can be defined on top of the existing libraries and components.

Language Interfaces on the Web

http://lionweb.io
info@lionweb.io

ACTIVITIES

1. The main focus is the **definition of protocols** for the communication between participating software components such as repositories and clients. To make this feasible, we also define a meta-metamodel as well as a reference architecture.

Language Interfaces on the Web

http://lionweb.io
info@lionweb.io

ACTIVITIES

1. The main focus is the **definition of protocols** for the communication between participating software components such as repositories and clients. To make this feasible, we also define a meta-metamodel as well as a reference architecture.

2. Where appropriate, in order to support the protocols, LIonWeb defines **programming language-level APIs** to access models and metamodels and to encapsulate the protocols.

Language Interfaces on the Web

http://lionweb.io
info@lionweb.io

ACTIVITIES

1. The main focus is the **definition of protocols** for the communication between participating software components such as repositories and clients. To make this feasible, we also define a meta-metamodel as well as a reference architecture.

2. Where appropriate, in order to support the protocols, LIonWeb defines **programming language-level APIs** to access models and metamodels and to encapsulate the protocols.

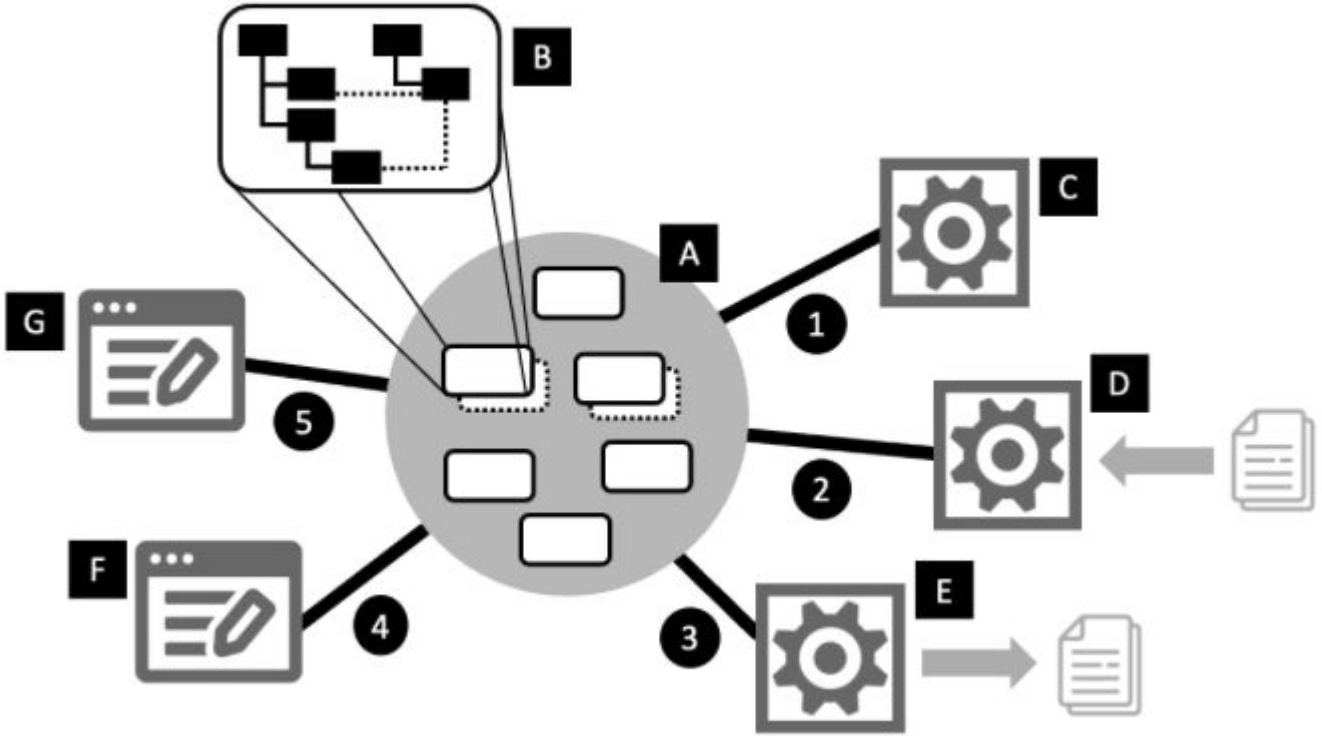
3. A third activity is to create a **collaboration hub** for the developers of such components, to empower other software developers to develop web-based modeling solutions, and to educate the public about the benefits of this approach.

LIONWEB

Language Interfaces on the Web

REFERENCE ARCHITECTURE

"LSP for models"

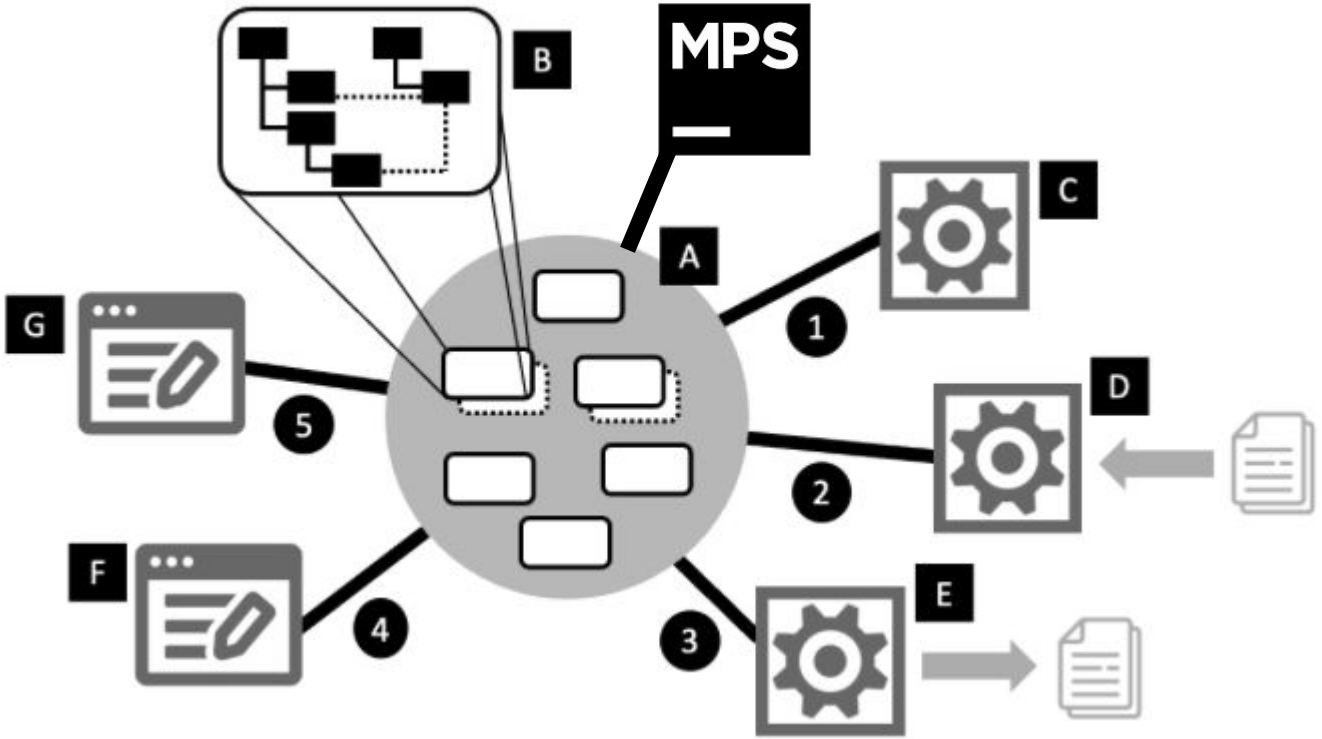


LIONWEB

Language Interfaces on the Web

REFERENCE ARCHITECTURE

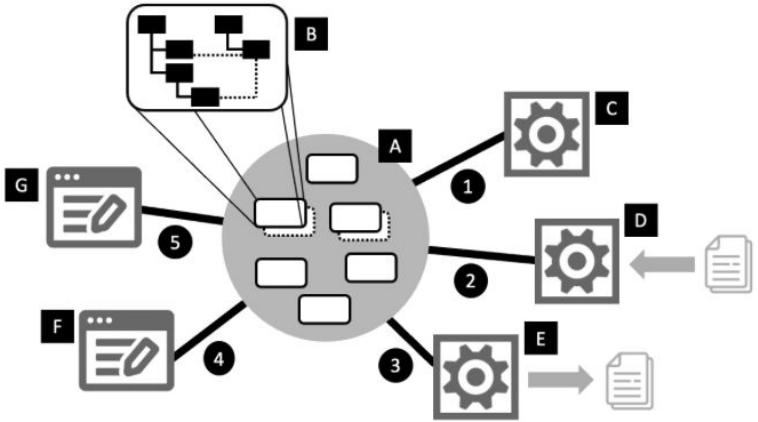
"LSP for models"



Language Interfaces on the Web

REFERENCE ARCHITECTURE

"LSP for models"



Bulk Access

Load a part of the model, modify, store it back
Load a part of the model, derive something else

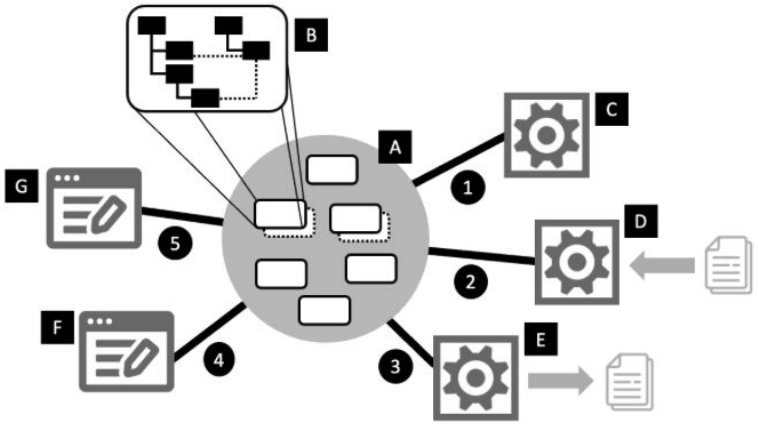
Delta Access

Initialise with a part of the model, get notified of changes, write back changes continuously

Language Interfaces on the Web

REFERENCE ARCHITECTURE

"LSP for models"



Primary Models

Edited by the user (ASTs)

Derived Models

Produced by processors based on other models; associated with primary models

Bulk Access

Load a part of the model, modify, store it back

Load a part of the model, derive something else

Delta Access

Initialise with a part of the model, get notified of changes, write back changes continuously

Language Interfaces on the Web

WORK PRODUCTS

Specifications for the communication protocols; this also requires the definition of a meta-metamodel

All our specifications and code is released as **open source** under the Apache 2.0. It explicitly allows commercial use. LIonWeb is currently not its own legal entity, it's just a voluntary collaboration of people.

LIONWEB

Language Interfaces on the Web

WORK PRODUCTS

Specifications for the communication protocols; this also requires the definition of a meta-metamodel

A reference architecture for cloud-based modeling tools (necessarily mainly to be able to produce the others)

All our specifications and code is released as **open source** under the Apache 2.0. It explicitly allows commercial use. LIonWeb is currently not its own legal entity, it's just a voluntary collaboration of people.

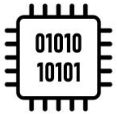
LIONWEB

Language Interfaces on the Web

WORK PRODUCTS

Specifications for the communication protocols; this also requires the definition of a meta-metamodel

A reference architecture for cloud-based modeling tools (necessarily mainly to be able to produce the others)



Where appropriate, bindings for these protocols in several programming languages

All our specifications and code is released as **open source** under the Apache 2.0. It explicitly allows commercial use. LIonWeb is currently not its own legal entity, it's just a voluntary collaboration of people.

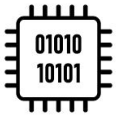
LIONWEB

Language Interfaces on the Web

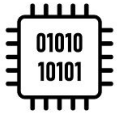
WORK PRODUCTS

Specifications for the communication protocols; this also requires the definition of a meta-metamodel

A reference architecture for cloud-based modeling tools (necessarily mainly to be able to produce the others)



Where appropriate, bindings for these protocols in several programming languages



Reference implementations and examples of how to build systems using the LIonWeb protocols (in particular, using some of the systems mentioned in the History section)

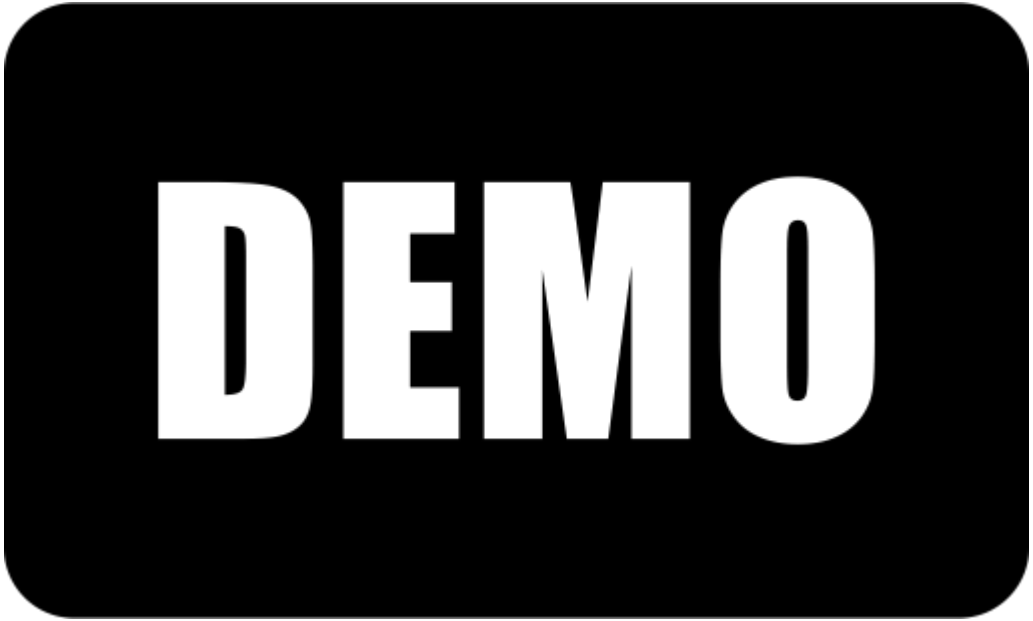
All our specifications and code is released as **open source** under the Apache 2.0. It explicitly allows commercial use. LIonWeb is currently not its own legal entity, it's just a voluntary collaboration of people.

Language Interfaces on the Web

PARTICIPATION

- Meinte Boersma, Freelancer, The Netherlands
- Norman Koester, itemis (modelix Team), Germany
- Sergej Koscejev, Freelancer, Czech Republic
- Sascha Lisson, itemis (modelix Team), Germany
- Eugen Schindler, Canon Production Printing, The Netherlands
- Alex Shatalin, Jetbrains (MPS Team), Czech Republic
- Niko Stotz, F1RE, The Netherlands
- Federico Tomassetti, Strumenta, Italy
- Markus Voelter, Freelancer, Germany
- Jos Warmer, openmodeling (Freon Team), The Netherlands

Language Interfaces on the Web



Language Interfaces on the Web

DEMO CONTENTS

1. Generate LIonWeb metamodel from Kotlin classes
2. Parse properties file and store as LIonWeb instance model
3. Import metamodel into MPS
4. Convert to MPS languages
5. Import properties instance model into MPS
6. Show instance model in Freon web editor, served from MPS via LIonWeb protocol
7. Change some values in Freon
8. Store changes back to MPS
9. Export example instance from MPS to LIonWeb
10. Unparse example instance

Language Interfaces on the Web

EXISTING PROTOCOLS

Bulk Read

Bulk Store

EXISTING CODE

APIs in Typescript

APIs in Java/Kotlin

(APIs in C#)

Integration with MPS

Integration with Freon

Integration with Modelix

Integration with StarLasu

Integration with EMF

LIONWEB

Language Interfaces on the Web

NEAR TERM FOCUS

- Finalize Bulk Protocols and document them
- Start Work on delta protocols / collaboration
- Refine meta-metamodel
- Improve reference implementations and examples
- Support integration with more tools

If you want to join and work with us, talk to any of the current members.
Or contact us all: info@lionweb.io

If you just want to stay in touch and use our stuff, please join our Slack:

https://join.slack.com/t/lionweb/shared_invite/zt-1uvaly9eb-z529c694OIN5oBh9FH1vhQ