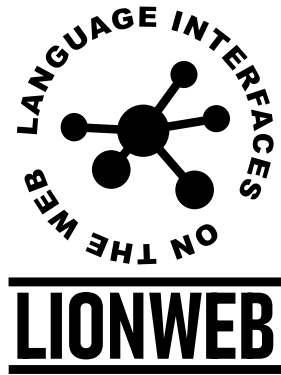# The LionWeb Initiative

at LangDev 2023
Jos Warmer, Niko Stotz

Project Overview
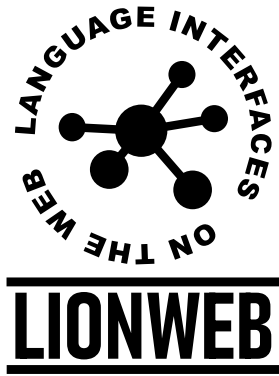
Mission

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

- Adopt language engineering
- Reduce vendor lock-in
- Mix and match components
- Foster innovation and incremental improvements

LIONWEB
LANGUAGE INTERFACES ON THE WEB

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

Why – Rationale

Most mature LWBs desktop-based

⇨ Everything needs to be web
⇨ Huge re-engineering effort

Language Engineering landscape is fragmented

⇨ Strong tool lock-in
⇨ Many interesting tools/component that you would like to use
   … but … you cannot
⇨ Everybody has to keep reinventing the wheel

Our community is too small to afford this, we can only grow if we can benefit from each others work

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

LionWeb: Language ⎰ Interfaces
                    ⎱ Interoperability ⎰ on the Web
                      Integration
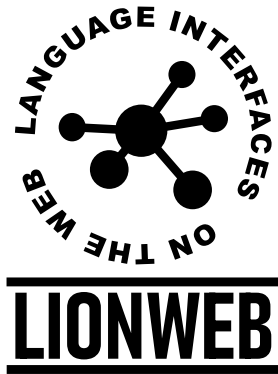
How – Principles

LionWeb as boring, reliable base

⇨ Innovate on top
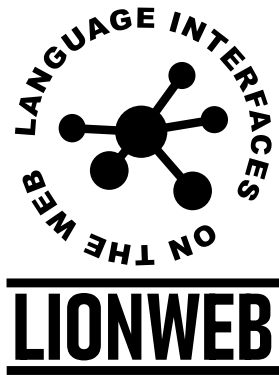
Rough consensus & Running code

⇨ Yoyo approach

Everything is a model

⇨ Markus Voelter's Whitepaper

"*A Platform for Systems and Business Modeling*"

Business-friendly Open Source

⇨ Apache 2.0 license

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

What are we working on

LANGUAGE INTERFACES ON THE WEB

LIONWEB

Interoperability

⇨ meta-metamodel
⇨ serialization format

2023.1 Release

⇨ protocols

Next release

- Programming APIs
  ⇨ Java
  ⇨ Kotlin
  ⇨ TypeScript
  ⇨ JavaScript

- Tool integrations
  ⇨ MPS
  ⇨ Starlasu
  ⇨ Freon
  ⇨ Ecore

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

What not – Out of Scope

- Yet another Language Workbench
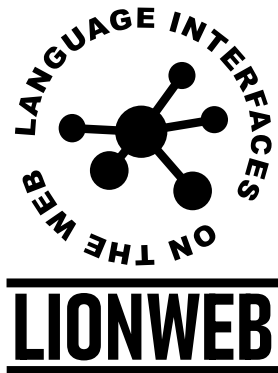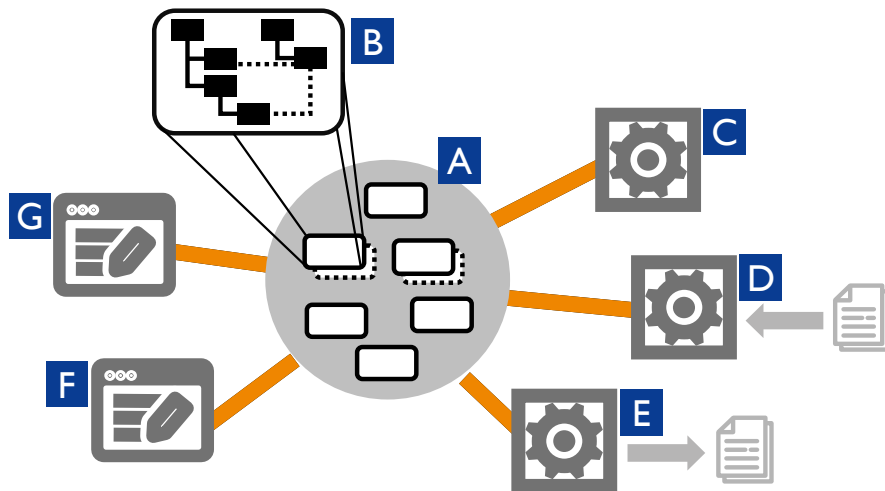

- Plug-and-play component system

- Deployment scheme

Technical Overview

Reference Architecture: Parts

Repository

Models
- **A** Original
- **B** Derived

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

JSON & protocols

Clients

Editors
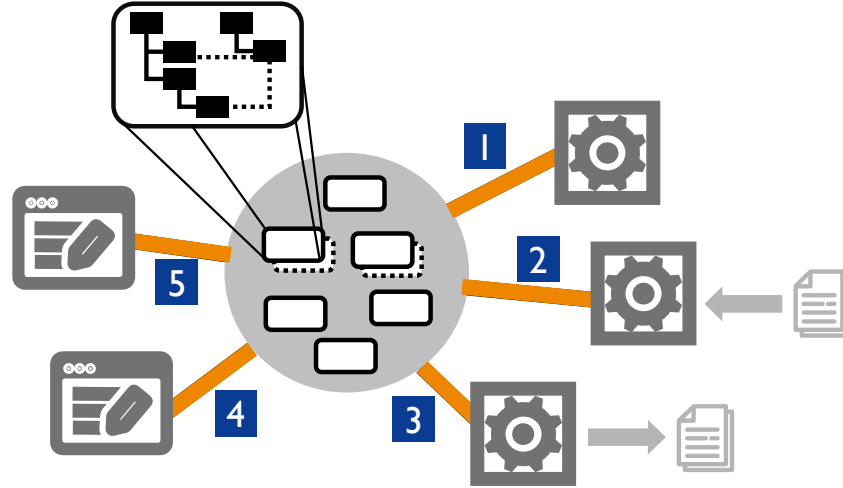- **F** Visual
- **G** Projectional

Processors
- **C** Model checker
- **D** Importer
- **E** Generator

Reference Architecture: Protocols

Bulk

2 **5** Load model, modify, store it back
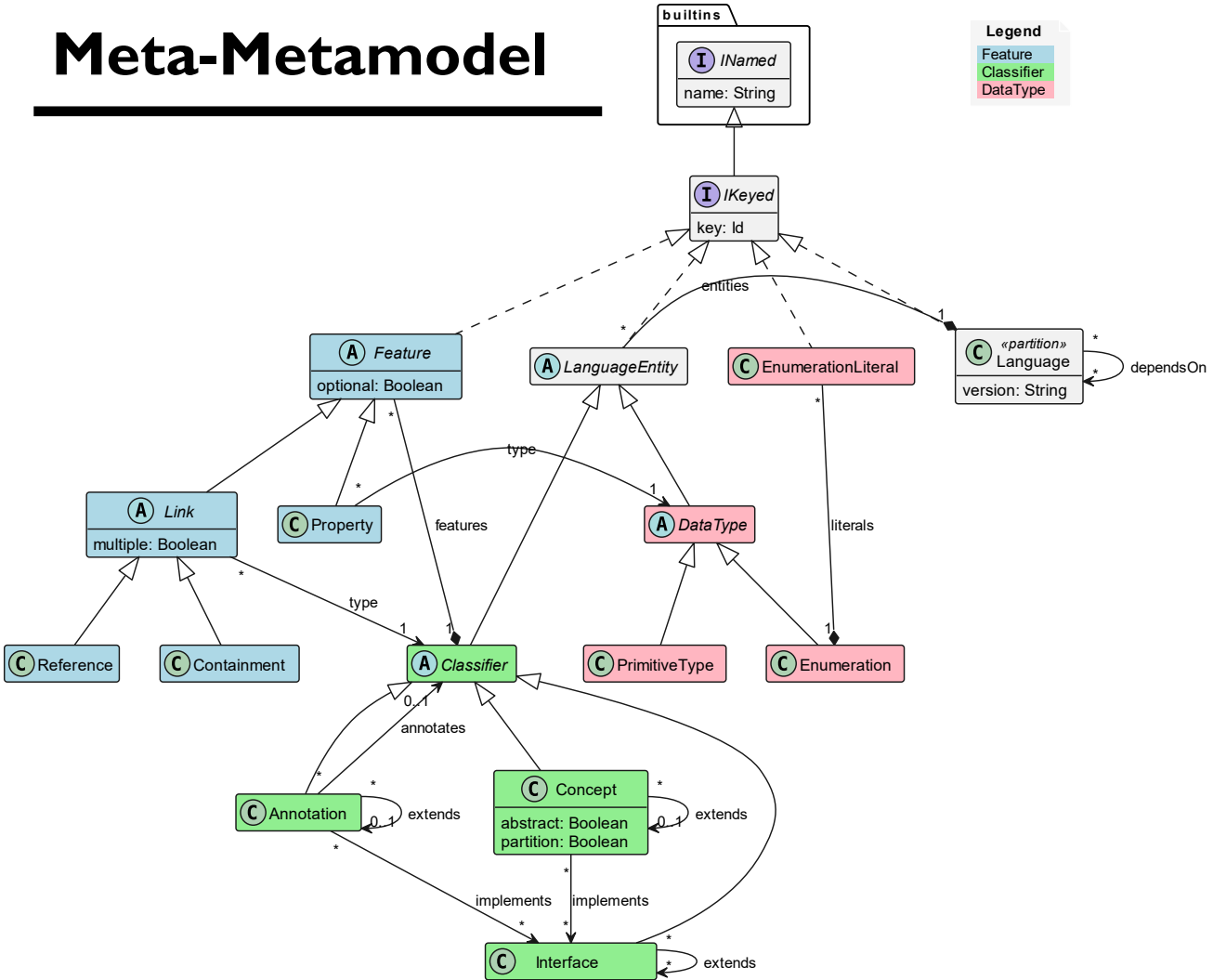
3 Load, derive something else

Delta

1 **4** Initialise with model, get notified of changes, write back changes continuously
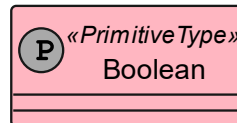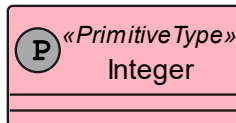
Meta-Metamodel

Built-in Meta Model Standard Library

«Concept»
Node
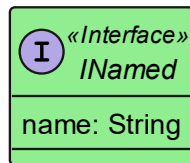
«Interface»
INamed

name: String

«PrimitiveType»
String

«PrimitiveType»
Integer

«PrimitiveType»
Boolean

«PrimitiveType»
JSON

LANGUAGE INTERFACES ON THE WEB
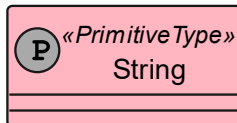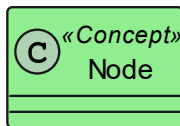
LIONWEB

Showtime

Demo available at https://github.com/LionWeb-io/property-demo

Example Metamodel: PROPS Language

Language Engineer Flow

Using MPS to create PROPS language

LionWeb PROPS metamodel

Kotlin Converter → Kotlin PROPS classes → Kotlin PROPS Application

MPS Converter → MPS PROPS Language → MPS with PROPS language

Freon Converter → Freon PROPS language → Freon PROPS Web Editor

Ecore Converter → Ecore PROPS language → Ecore PROPS language + Xtend

End User Flow

Properties file

1

Kotlin PROPS
Application

8

Changed
Properties file

2

PROPS LionWeb
Instance.json

3

7

MPS with PROPS
language

5

EMF PROPS
instance

6

HTML Page
with PROPS
instance

Freon PROPS
Web Editor

4

LionWeb Team	
Meinte Boersma	Freelancer
Norman Koester	itemis (modelix)
Sergej Koscejev	Freelancer
Sascha Lisson	itemis (modelix)
Eugen Schindler	Canon Production Printing
Alex Shatalin	Jetbrains (MPS)
Niko Stotz	FIRE
Federico Tomassetti	Strumenta (Starlasu)
Markus Voelter	Freelancer
Jos Warmer	Freelancer (Freon)

Want to join and work with us:
info@lionweb.io

Want to follow us or use or *implement* LionWeb:
https://join.slack.com/t/lionweb/shared_invite/zt-1uvaly9eb-z529c694OIN5oBh9FH1vhQ

LANGUAGE INTERFACES ON THE WEB

JOIN US