LionWeb: Status + Delta Protocol

LangDev 2025

Meinte Boersma, Niko Stotz, Ulyana Tikhonova

Status

Mission

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

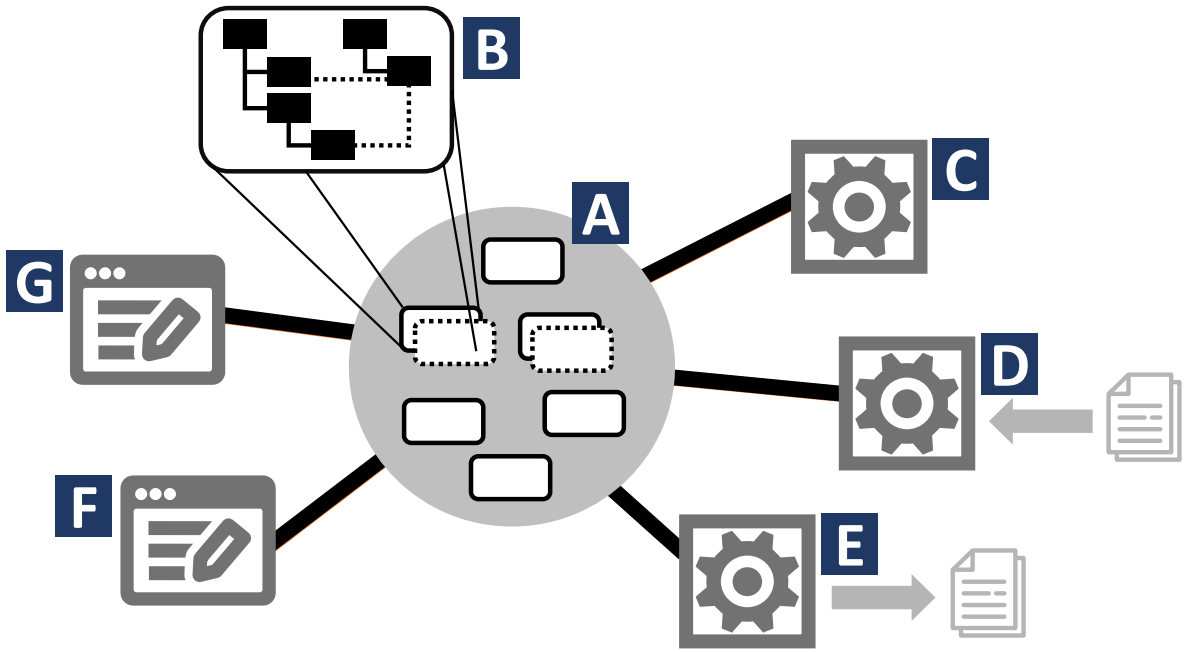
Adopt language engineering

Reduce vendor lock-in

Mix and match components

Foster innovation and incremental improvements

Reference Architecture



Repository

Models

- A Original ✓
- B Derived ✓

Protocols (bulk ✓ / delta ✓)
Contents (JSON ✓ / Protobuf ✓)

Clients

Editors

- F Visual
- G Projectional ✓

Processors

- C Model checker
- D Importer ✓
- E Generator ✓

Talks

- LangDev 2023 (slides, video)
- LangDev 2024 (slides, video)
- SEN Symposium 2024 (conference, slides)
- MODELS 2024 (conference)
- MBSE 2025 (conference)

Working documents

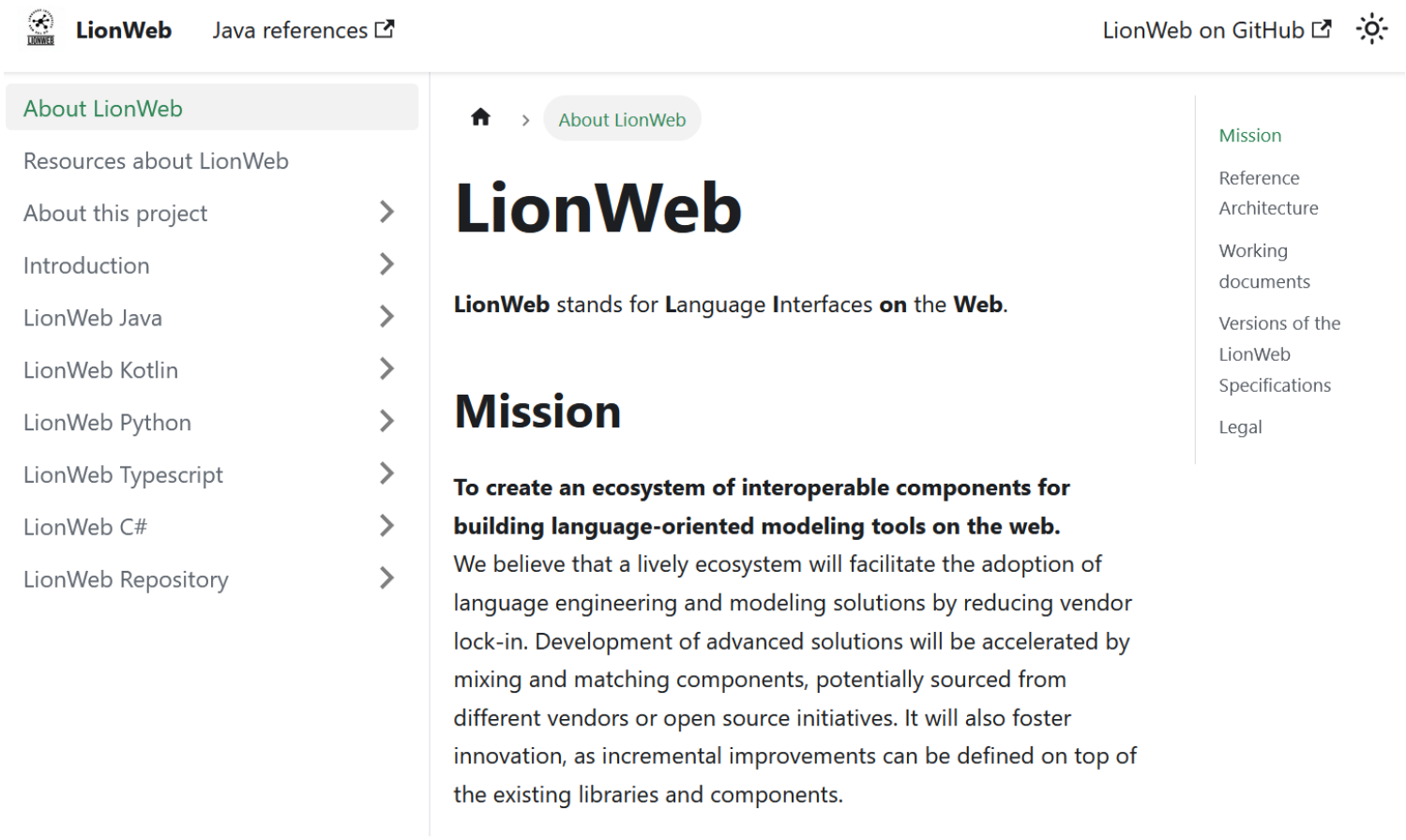
- Roadmap
- Reference architecture
- Use cases

Other talks about LionWeb @ LangDev 25

Title	Speaker	Affiliation
Protostar: an AST solution for ANTLR4 and LionWeb Thursday 9:30	Alessio Stalla	Strumenta
LionWeb's delta protocol: incremental model changes Thursday 15:00	Meinte Boersma, Niko Stotz	DSL Consultancy, FIRE
Making DSLs conversable Thursday 15:30	Krishna Narasimhan	FIRE
Freon-Powered: Crafting Web-Native DSL Editors That Wow Thursday 17:00	Jos Warmer	OpenModeling
SysML v2 language in LionWeb Friday 10:10	Ulyana Tikhonova	FIRE

Better documentation

lionweb.io



Better outreach

New core members

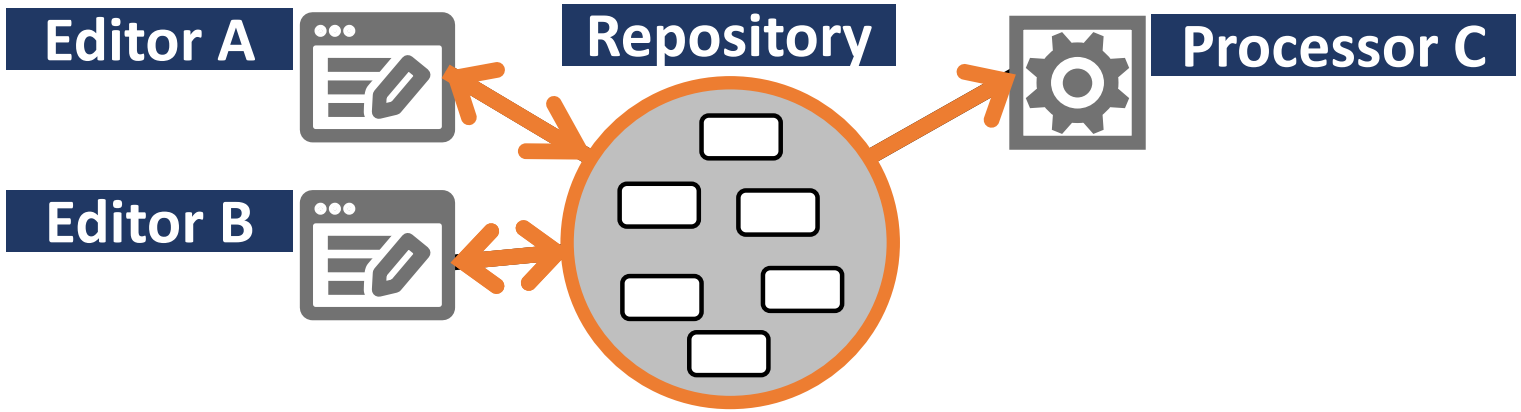
- Pedro Molina / Metadev
- Ulyana Tikhonova, Erkan Diken / FIRE

Office hours

- Once a month (every second Friday)

Delta protocol

Basic idea



Assumption: Every client starts from same model state

1. Editor A sends command `ChangeProperty` to Repository
2. Repository checks and integrates command
3. Repository sends event `PropertyChanged` to all clients
4. Editor B sends query `SignOn` to Repository
5. Editor B sends query `SubscribeToPartitionContents` to Repository, receives partition contents

Message kinds

Command	Client ⇨ Repository	Async	“I want to change this”
Event	Client ⇦ Repository	Async	“This changed”
Query	Client ⇔ Repository	Synchronous	“Tell me about this”

All commands

Action		Child	Annotation	Reference	Property	Partition / Classifier
Add		Add child	Add annotation	Add reference Add reference ResolveInfo Add reference target	Add property	Add partition
Delete		Delete child	Delete annotation	Delete reference Delete reference ResolveInfo Delete reference target	Delete property	Delete partition
Change		Replace child	Replace annotation	Change reference Change reference ResolveInfo Change reference target	Change property	Change classifier
Move	from other node	Move child from other containment	Move annotation from other parent	Move entry from other reference		
	from other feature in same node	Move child from other containment in same parent		Move entry from other reference in same parent		
	in same feature in same node	Move child in same containment	Move annotation in same parent	Move entry in same reference		
Move and replace existing	from other node	Move child from other containment and replace existing child	Move annotation from other parent and replace existing annotation	Move entry from other reference and replace existing entry		
	from other feature in same node	Move child from other containment in same parent and replace existing child		Move entry from other reference in same parent and replace existing entry		
	in same feature in same node	Move child in same containment and replace existing child	Move annotation in same parent and replace existing annotation	Move entry in same reference and replace existing entry		
Composite		Composite				

All queries

Subscription

- Subscribe to partition changes
- Subscribe to partition contents
- Unsubscribe from partition contents

Participation

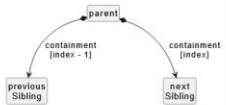
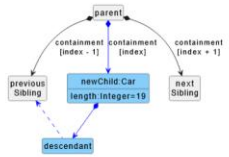
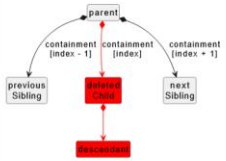
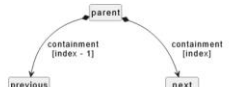
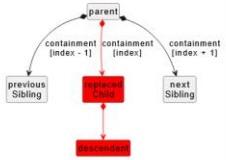
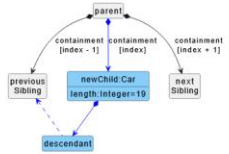
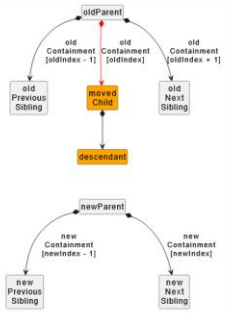
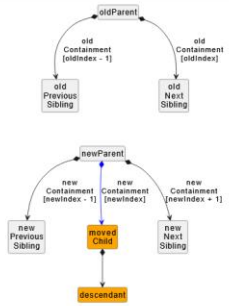
- Sign On
- Sign Off
- Reconnect

Miscellaneous

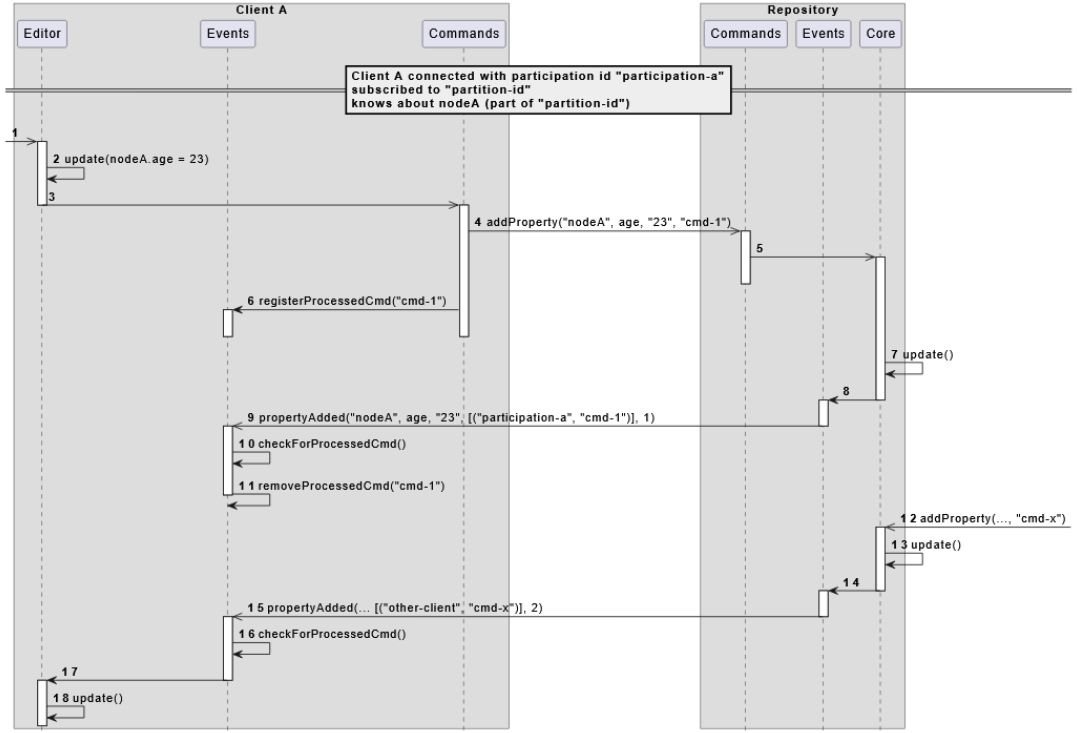
- Get available ids
- List partitions

Well documented

6.4.4. Children

Command / Event / Description	Before	After
Add child Child added Add new node `newChild` to `parent` in `containment` at `index`.		
Delete child Child deleted Delete existing node `deletedChild` from `parent`'s `containment` at `index`.		
Replace child Child replaced Replace existing node `replacedChild` inside `parent`'s `containment` at `index` with new node `newChild`. Delete `replacedChild`.		
Move child from other containment Child moved from other containment Move existing node `movedChild` (currently inside `oldParent`'s `oldContainment` at `oldIndex`) inside `newParent`'s `newContainment` at `newIndex`.		

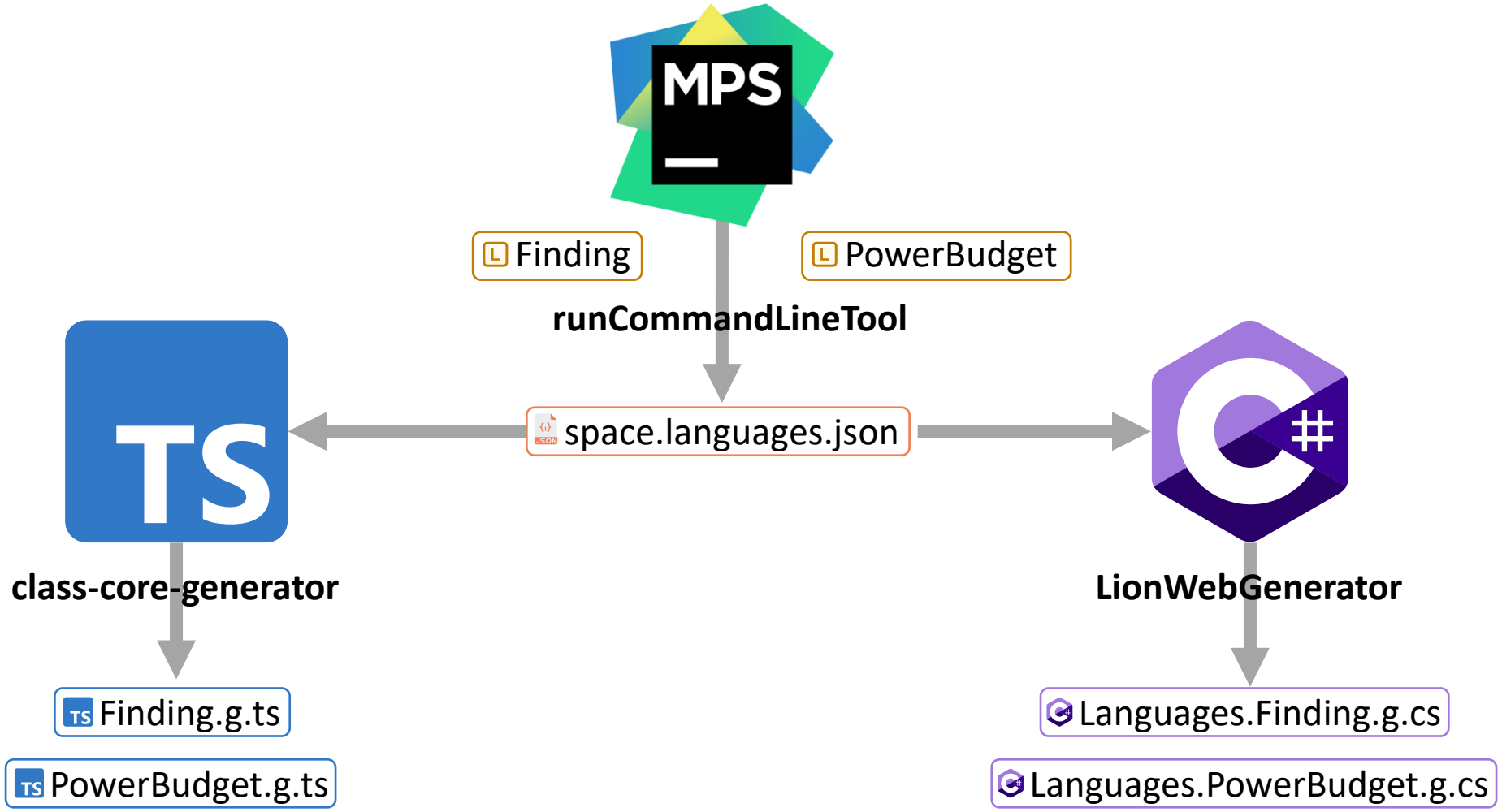
5.4. Client changes known node with local update



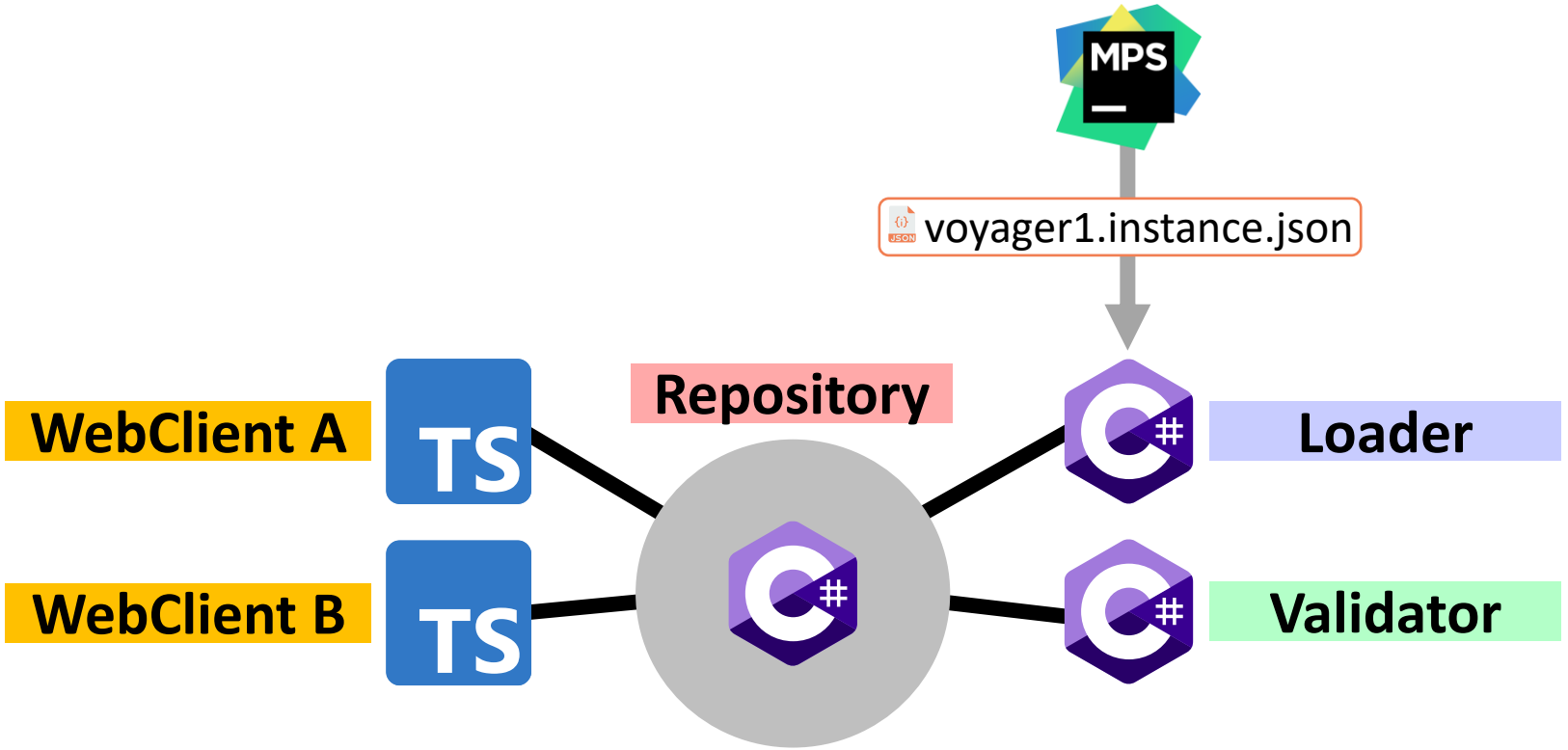
Demo

https://github.com/LionWeb-io/space-demo

Languages (M2)



Instances (MI)



Team and Contact

Team

Meinte Boersma	DSL Consultancy
Erkan Diken	FIRE
Norman Koester	itemis (Modelix)
Sergej Koscejev	Freelancer
Sascha Lisson	itemis (Modelix)
Pedro Molina	Metadev
Eugen Schindler	Canon Production Printing
Alex Shatalin	JetBrains (MPS)
Niko Stotz	FIRE
Federico Tomassetti	Strumenta
Markus Voelter	Freelancer
Ulyana Tikhonova	FIRE
Jos Warmer	Freelancer (Freon)

Contact

Want to work with us:
info@lionweb.io

GitHub:
https://github.com/LionWeb-io

Web:
https://lionweb.io/

Slack to follow us, use, or implement LionWeb:
https://join.slack.com/t/lionweb/shared_invite/zt-2k1k5bsv7-EvC2IABIgSxWrqJNVOZ0HQ