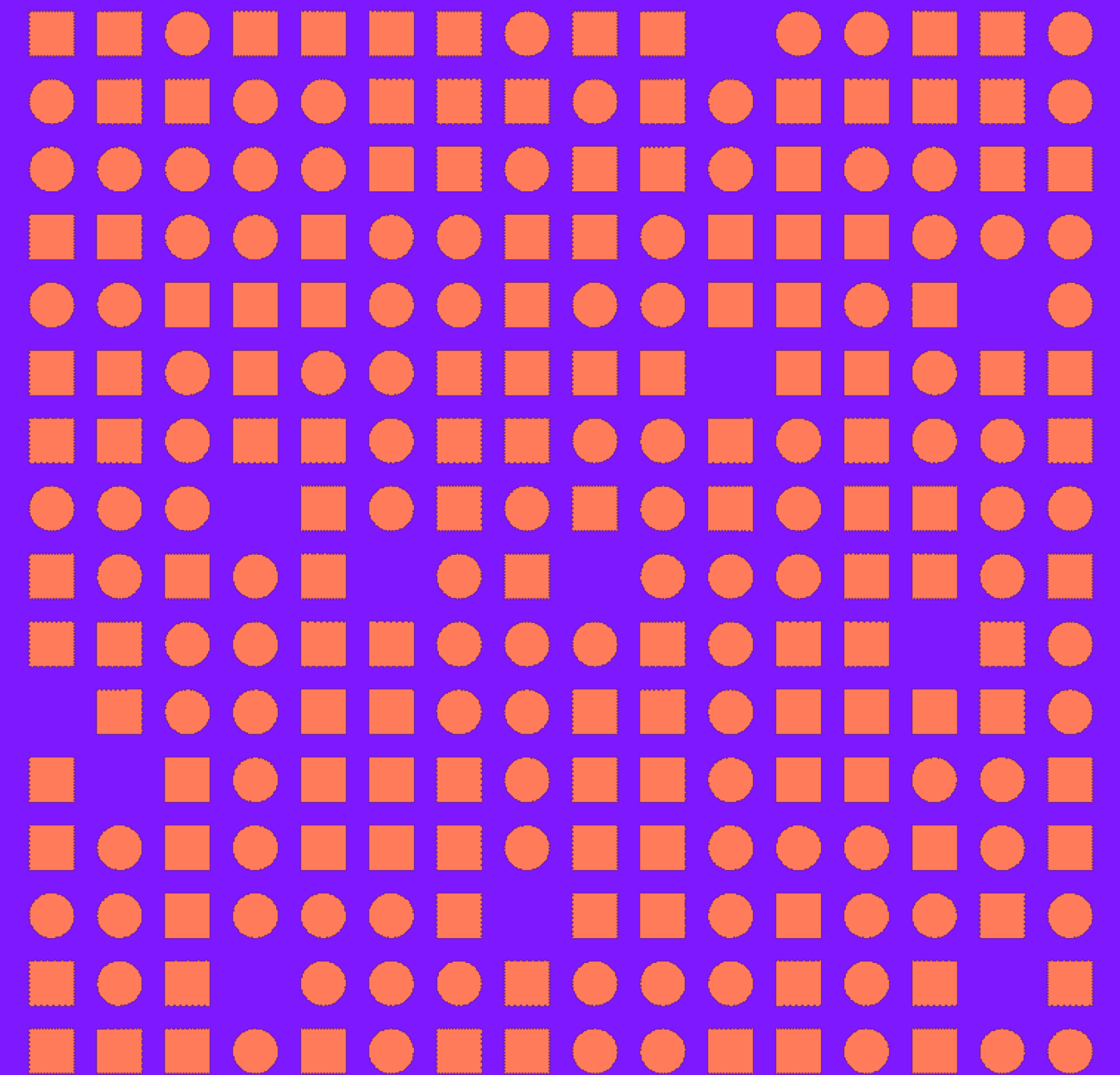


TypeFox



Introducing Typir: Type Checking for the Web!

Johannes Meier



About TypeFox



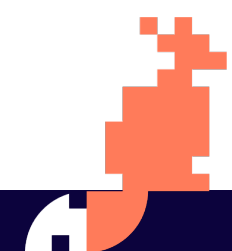
Founded in 2016

HQ in Kiel, Germany

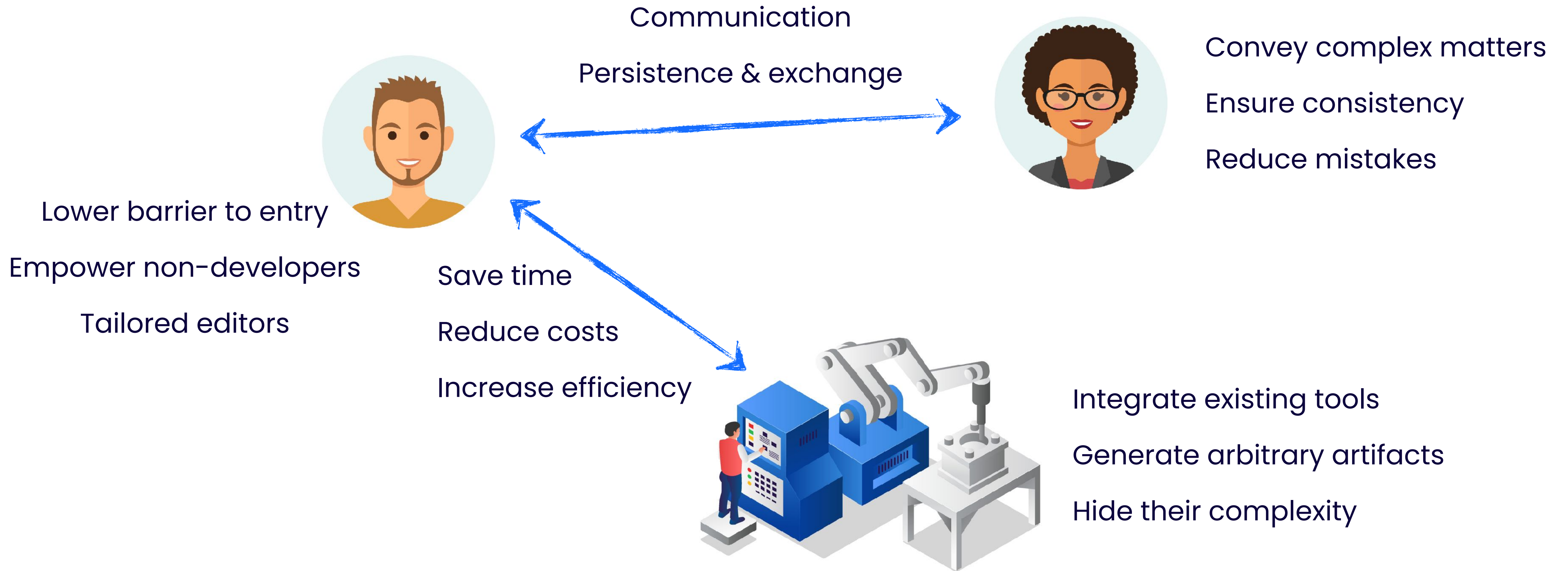
Open source frameworks for ...

- IDEs  **THEIA**
- DSLs  **Langium**

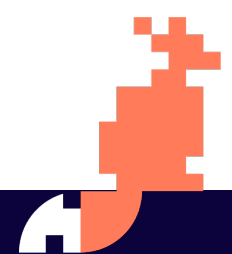
Custom-tailored solutions



Motivation for DSLs



Images by Vecteezy.com



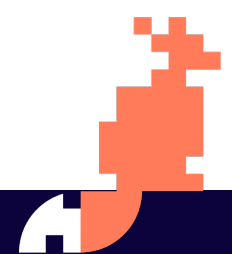
Checks in language engineering

1. Parser errors
2. Linking errors
3. Language-specific validations:
 - Syntactic checks
 - Semantic checks at development time
 - Semantic checks at runtime



Type checking:

- Annotate AST with types
- Checks on these types

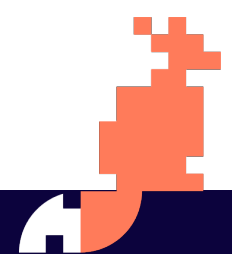


Motivation for type checking

- Validation of type errors
 - Type-related constraints
 - Assignability including sub-typing, casting, ...
- Resolving cross-references
- Generators
- ... with helpful error messages!

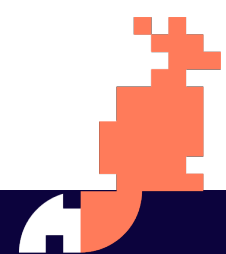


Live-Demo: LOX*



Motivation for Typir

- Support language engineering and modelling projects in practise
 - Easy starting point for type checking
 - Reuse for recurring patterns
- ... in the web!



Introducing Typir!

API for
Language
engineers

Library for type checking

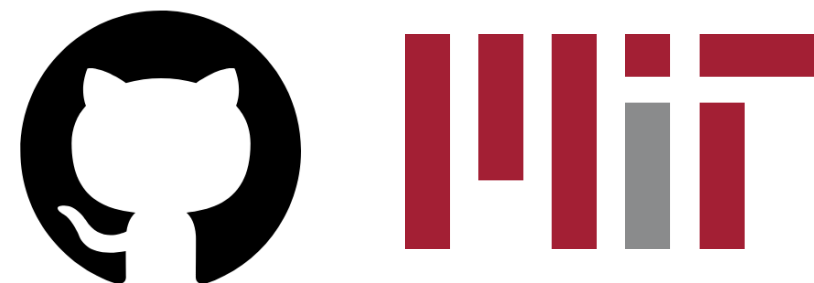


- ▶ Core features
- ▶ Reusable types

Typir



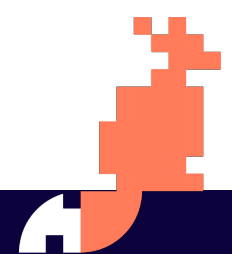
Open source



Pragmatic

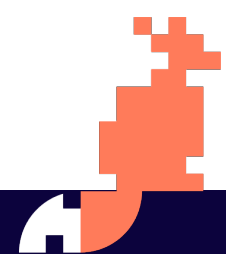
Early state

- ▶ since 2024
- ▶ by **TypeFox**



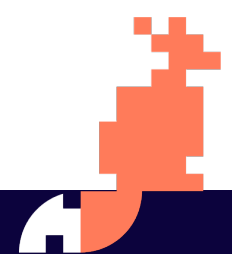
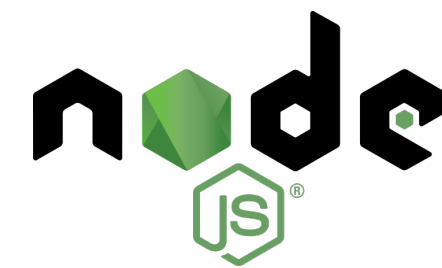
Pragmatic type checking

- More pragmatic than formal
- Default implementations for recurring problems
 - Kinds of types: primitives, classes, functions, ...
 - Algorithms: Circular type definitions, performance/caching, ...
 - Meaningful error messages
- Provided with TypeScript API



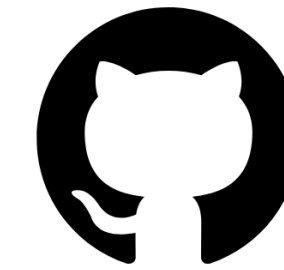
Ready for the web

- Written in TypeScript
- Runs in web browsers, e.g. web app, ...
- Runs in Node.js, e.g. web server, CLI, ...
- Runs in desktop applications, e.g. VS Code, Theia, ...



Open source

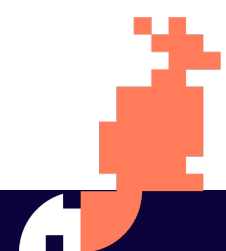
- Source code is open: <https://github.com/TypeFox/typir/>



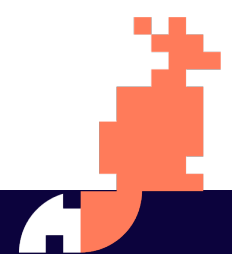
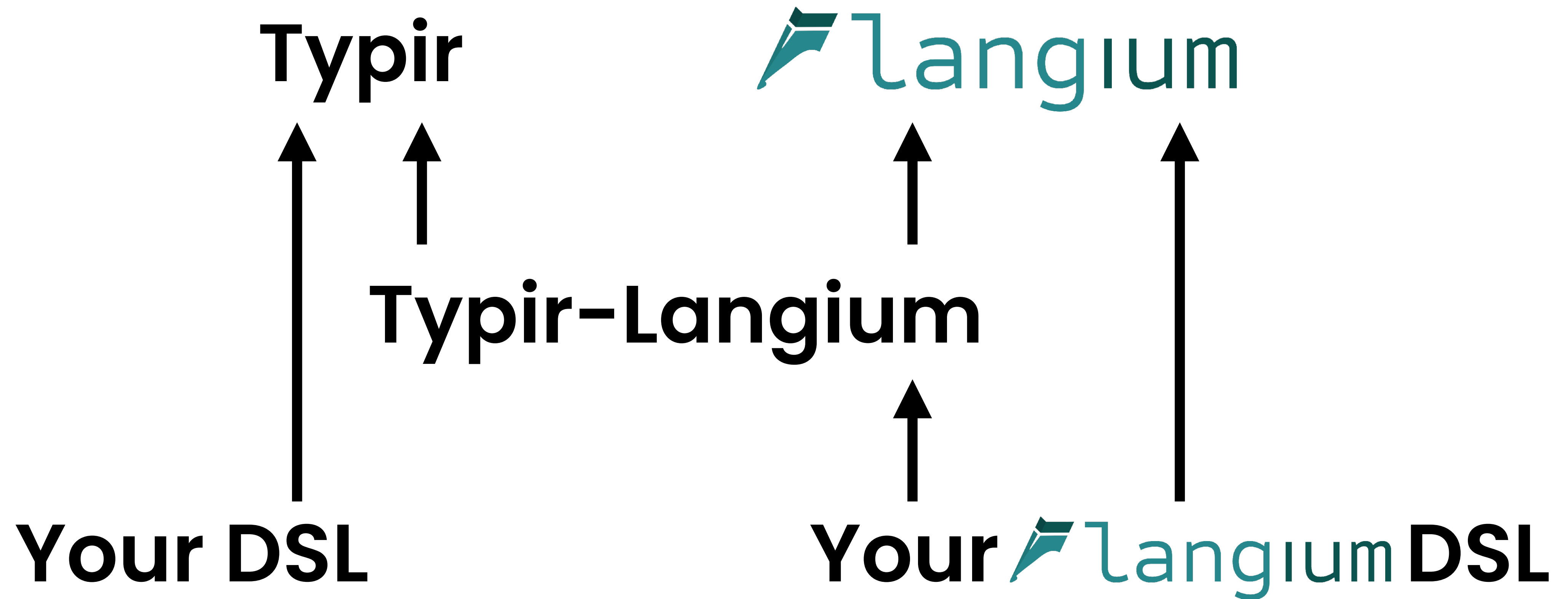
- MIT license



- ✓ Commercial projects
- ✓ Closed projects
- ✓ no fees, no contracts
- Collaboration with community
 - GitHub Discussions
 - We are open for contributions and funding!



What about Langium?



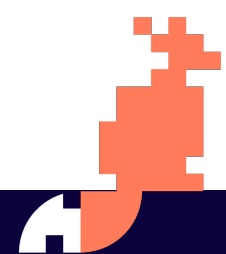
Customization

... by dependency injection:

- Services
- ... with default implementations
- Exchangeable implementations
- Parts of default implementations can be overridden

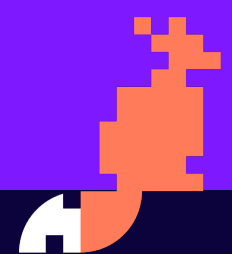
... by using custom types:

- Registry for kinds
- Type graph accepts any types



Live demo

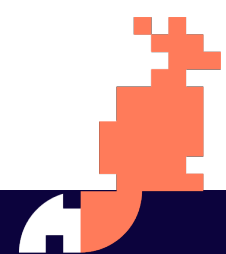
Applying Typir for LOX*



Core features

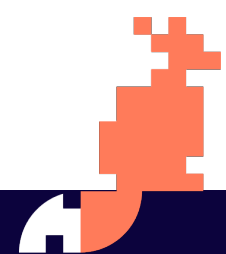
- Assignability: $\text{Type} \times \text{Type} \rightarrow \text{Boolean}$
- Subtype check: $\text{Type} \times \text{Type} \rightarrow \text{Boolean}$ (nominal vs. structural)
- Coercion/casting: $\text{Type} \times \text{Type} \rightarrow \{\text{Implicit}, \text{Explicit}, \text{None}, \text{Self}\}$
- Equality: $\text{Type} \times \text{Type} \rightarrow \text{Boolean}$

- Inference: $(\text{node: unknown}) \rightarrow \text{Type}$
 - Langium: $\text{AstNode} \rightarrow \text{Type}$
- Validation: $(\text{model: unknown}) \rightarrow \text{ValidationHints}$
 - Langium: $\text{AST} \rightarrow \text{ValidationHints}$



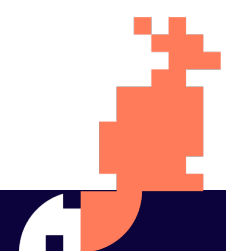
Supported types

- Primitive types
- Fixed parameter types, e.g. `List<T>`, `Map<K, V>`
- Functions, incl. overloading
- Classes: super-classes, fields and methods
- Top type (`any`)
- Bottom type (`never`)
- Operators (mapped to Functions)



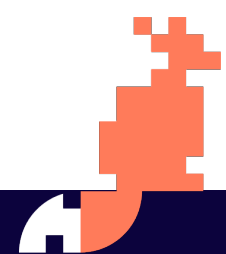
Roadmap

- Typir release v0.1.0 probably in November 2024
- Iterative development
 - Adding new features
 - Applying Typir to commercial projects
- Bindings for more frameworks: Sprotty, GLSP, LionWeb, ...



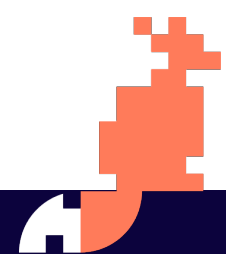
Features for the near future

- Support more types ...
 - Lambdas
 - Generic types
 - Enumeration types
- Scopes, type assertions
- Cyclic type definitions, e.g. `Node { children: Node[] }`
- Improve APIs regarding performance
- ...



Summary: Benefits of Typir

- Predefined types to reuse + customizations
- Builtin solutions for ...
 - Cyclic type definitions
 - Caching
 - Useful error messages for users of the DSL
- Bindings for language workbenches
- ... in the web!



TypeFox



Discussions & Ideas !

typir.org

typefox.io

*Meet us
at our booth!*

