Agile and Dependable Service Development

with Bosque and Morphir

Writing microservices should be easy

- They’re just functions:
  - Business logic that takes a request and returns an answer.

- A lot standards in our way if this simplicity:
  - Bugs
  - Testing
  - Technology and framework choices
  - Documentation
  - Hygiene & keeping tech up-to-date
  - Following blueprints and best practices
  - Regulations
  - Monitoring
  - Security & vulnerability remediation
  - Supportability
  - Audit
  - Evolution & future readiness
  - Telemetry, metrics, and observability

Technology To Reduce the Friction

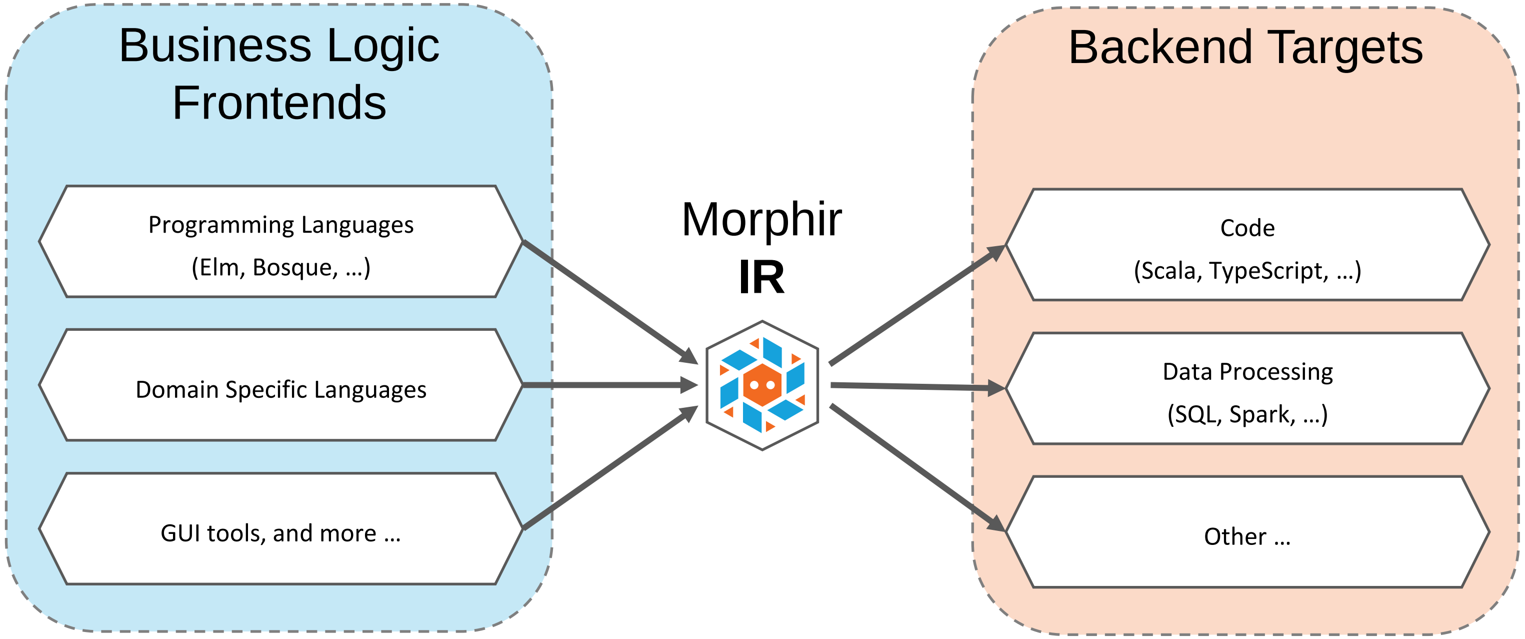
- Get developers back to focusing on the business logic and specs. Automate the rest:

- Bosque
  - Bugs
  - Testing
  - API
  - etc.

- Morphir
  - Optimize user/developer dialog
  - Boilerplate automation

Morphir

- **FINOS** project contributed by **Morgan Stanley**
- **Core**: Business-focused Intermediary Representation (IR)
- **Tools**: parsers, verifiers, **transpilers**


Business Logic Frontends
Programming Languages (Elm, Bosque, …)
Domain Specific Languages
GUI tools, and more …
Morphir IR
Backend Targets
Code (Scala, TypeScript, …)
Data Processing (SQL, Spark, …)
Other …

Business / Developer Collaboration

- Improve development by bringing the business and developer closer.
- Shared language
- Instant feedback

Example

- Let’s look at designing another service that uses our weather forecast.
- This time, we’re modeling decisions. In this case, we have a sailboat rental company and want to determine of rentals should be open based on the weather.
- This involves getting the decision specifications from the company owner.
- We get these decision points, which we review with Morphir tools:
  - Only novice and experts during high wind conditions
  - Only expert sailors during dangerous winds
  - No rentals during thunderstorms
- …

Using Morphir to Collaborate

- First off, how do we categorize windspeeds to the owner’s descriptions?