FINOS

Fintech
Open Source
Foundation

Sharing Business Logic
..across people &
technologies

morphir

What does Morphir do?

- **Shares business logic** across people and technologies

What is business logic?

- The app according business experts - in their language

Use Case: Surfboard Rentals

How many boards are available to rent?

Business View

$$\boldsymbol{available = current\ inventory - reserved + probable\ returns}$$

Where:

$$current\ inventory = \sum surfboards\ in\ store$$

$$reserved = \left\lfloor \left(\sum surfboard\ reservations \right) 0.9 \right\rfloor$$

$$probable\ returns = \frac{\sum scheduled\ returns}{2}$$

Technology View

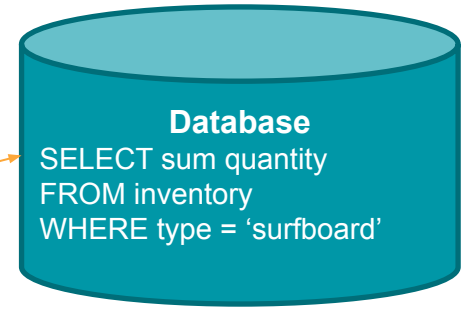
$$available = current\ inventory - reserved + probable\ returns$$

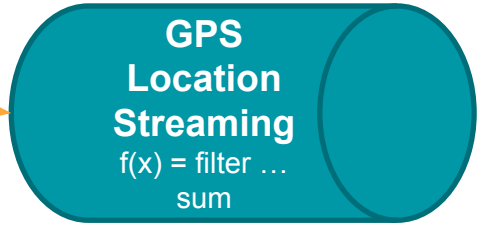
Where:

$$current\ inventory = \sum surfboards\ in\ store$$

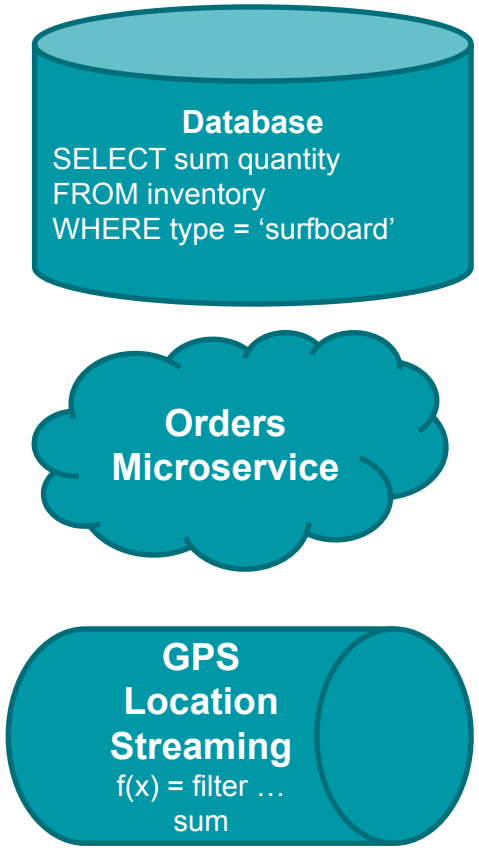
$$reserved = \left\lfloor \left(\sum surfboard\ reservations \right) 0.9 \right\rfloor$$

$$probable\ returns = \frac{\sum scheduled\ returns}{2}$$

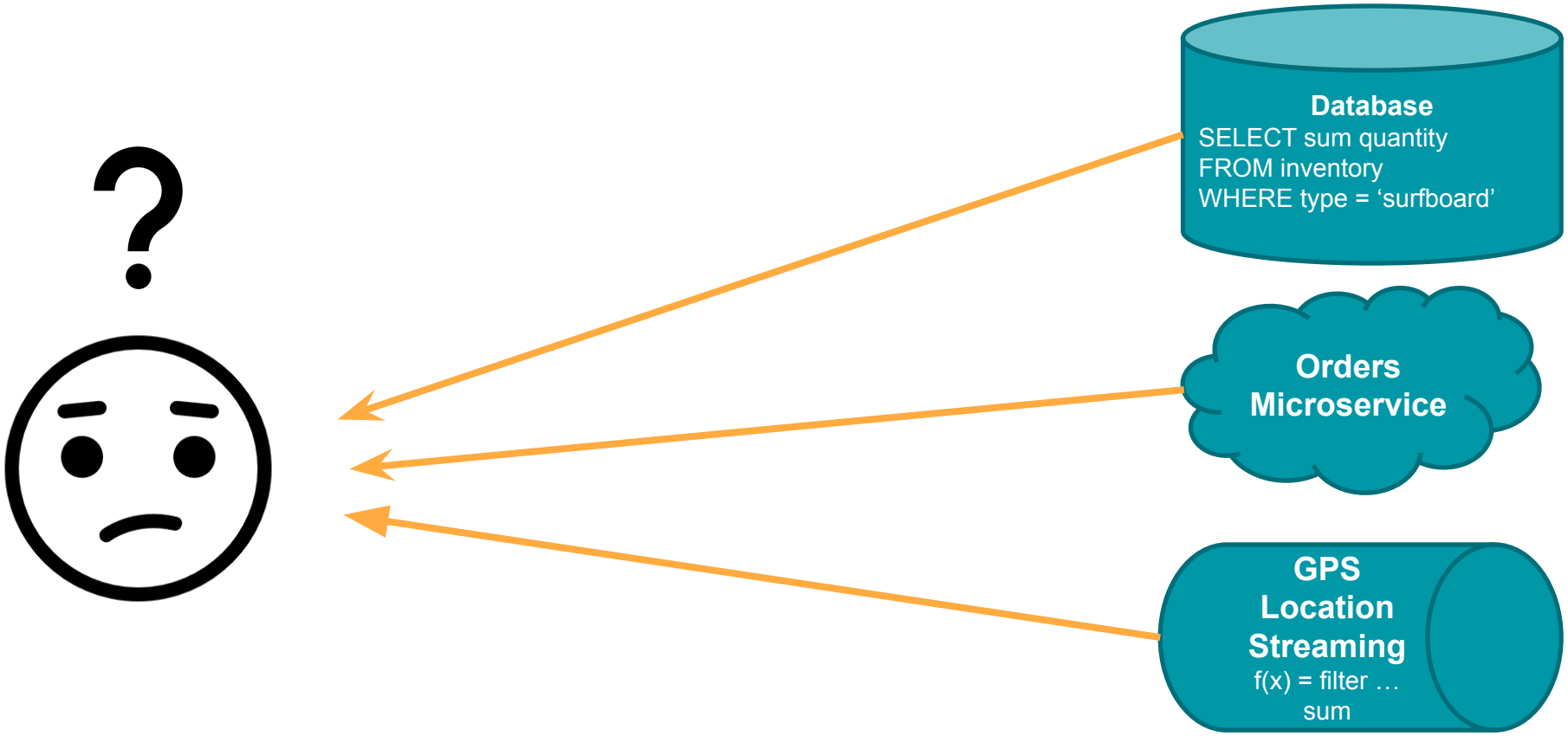




Evolved View



Scrambled View



Get back to…

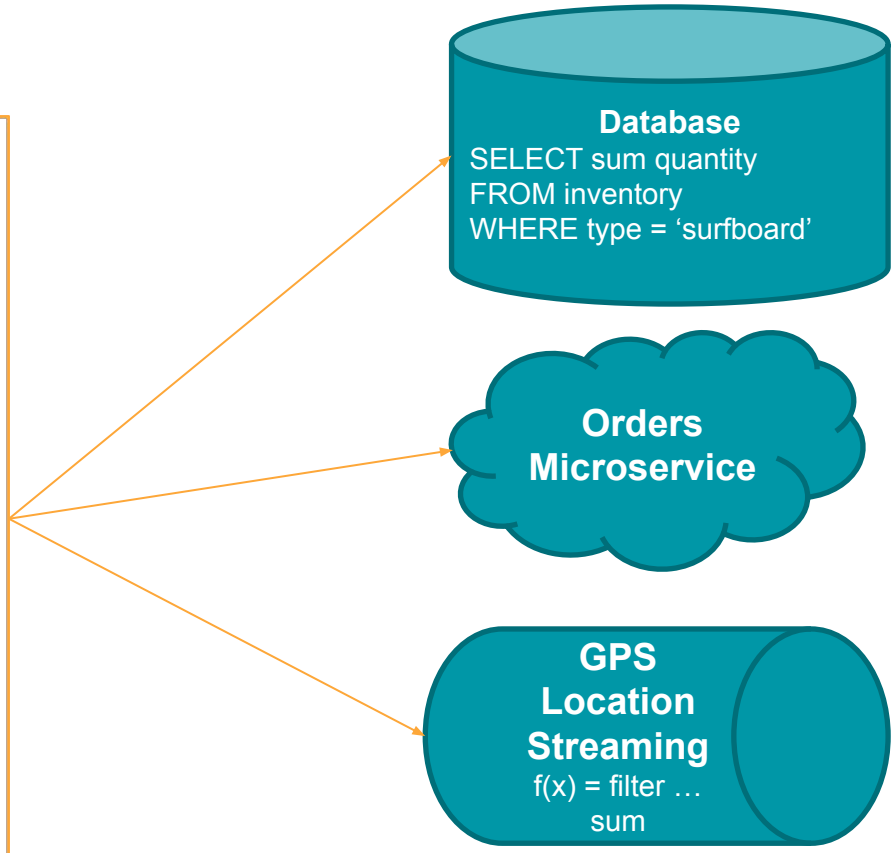
$$available = current\ inventory - reserved + probable\ returns$$

Where:

$$current\ inventory = \sum surfboards\ in\ store$$

$$reserved = \left\lfloor \left(\sum surfboard\ reservations \right) 0.9 \right\rfloor$$

$$probable\ returns = \frac{\sum scheduled\ returns}{2}$$



morphir

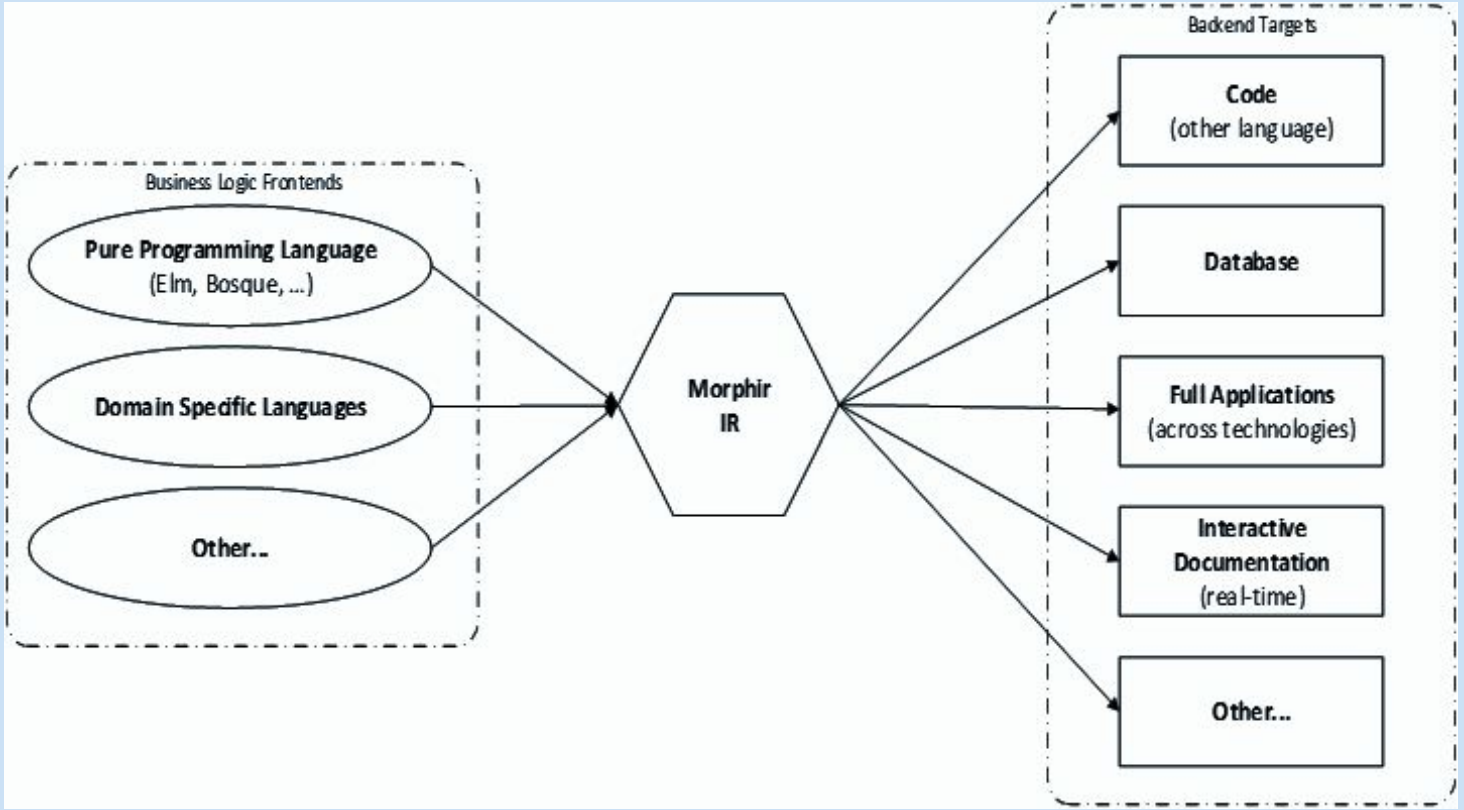
business logic interchange

morphir core: Intermediary Language (IR)

morphir tools: Parsers & Processors

quickly: Business □ Software

community: Ecosystem



Leads to:

Machine processable:

- Declarative
- Pure Business Logic
- Functional Programming

Enables:

- Fast feedback loop
- Verified safe programs
- Testing tools
- Replay & Audit
- Catalog & Lineage
- Language integration

Turning logic into data

- Morphir IR (intermediate representation)
- Functional Programming
  - Simple but powerful
  - Correctness
  - No side-effects

```
request allowPartial availableSurfboards requestedSurfboards =
    if availableSurfboards < requestedSurfboards then
        if allowPartial then
            Reserved (min availableSurfboards requestedSurfboards)

        else
            Rejected

    else
        Reserved requestedSurfboards
```

elm

parsed into

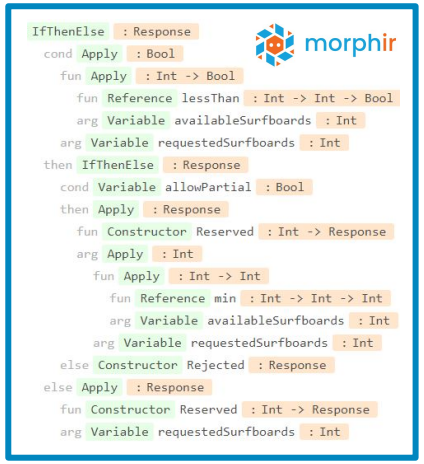
morphir

```
IfThenElse : Response
  cond Apply : Bool
    fun Apply : Int -> Bool
      fun Reference lessThan : Int -> Int -> Bool
      arg Variable availableSurfboards : Int
    arg Variable requestedSurfboards : Int
  then IfThenElse : Response
    cond Variable allowPartial : Bool
    then Apply : Response
      fun Constructor Reserved : Int -> Response
      arg Apply : Int
        fun Apply : Int -> Int
          fun Reference min : Int -> Int -> Int
          arg Variable availableSurfboards : Int
        arg Variable requestedSurfboards : Int
    else Constructor Rejected : Response
  else Apply : Response
    fun Constructor Reserved : Int -> Response
    arg Variable requestedSurfboards : Int
```

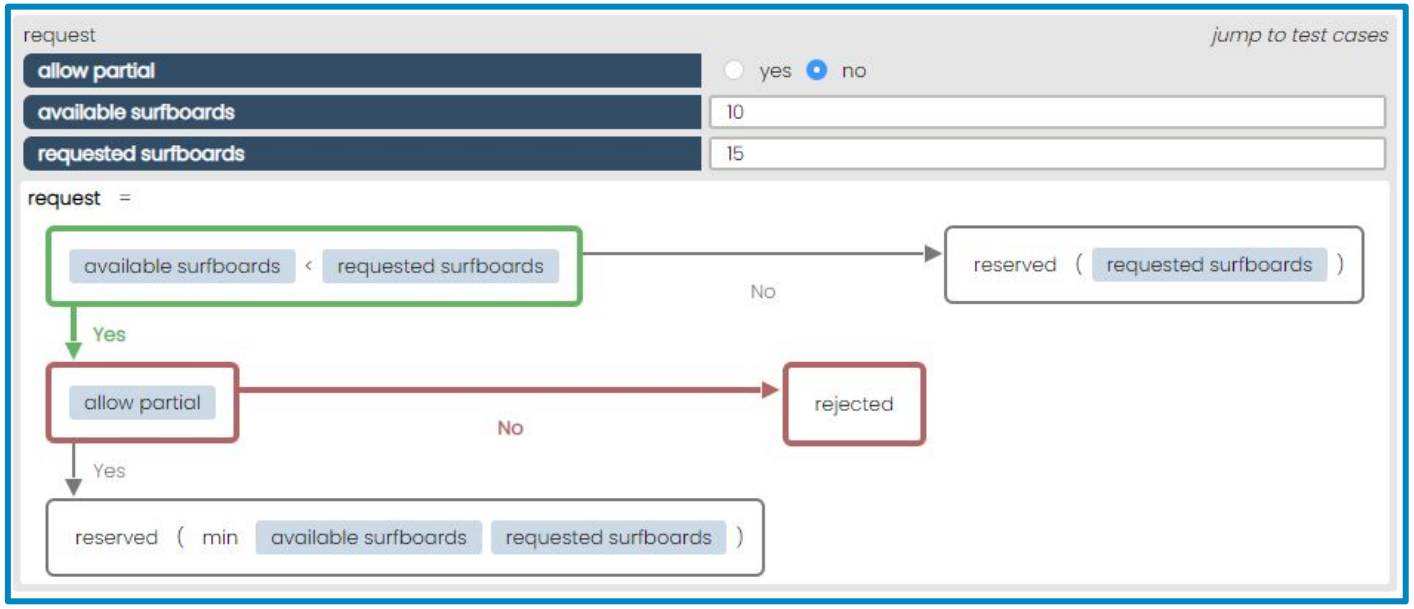
Visualizing the logic

- Visualizing logic is hard, but Morphir turns it into data
- Data driven UIs are easy
- Adapts to business logic
- Fully automated

- Tight feedback loop
- IT working more cooperatively with the business
- Live spec / documentation
- Production support



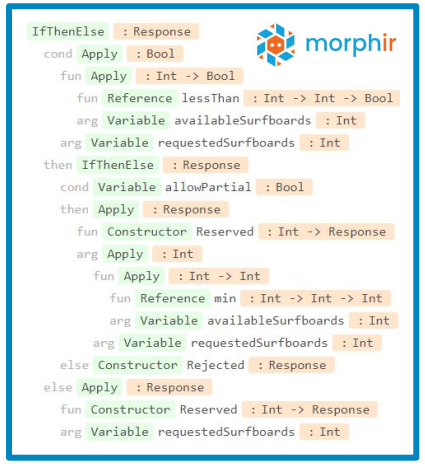
data driven visualization



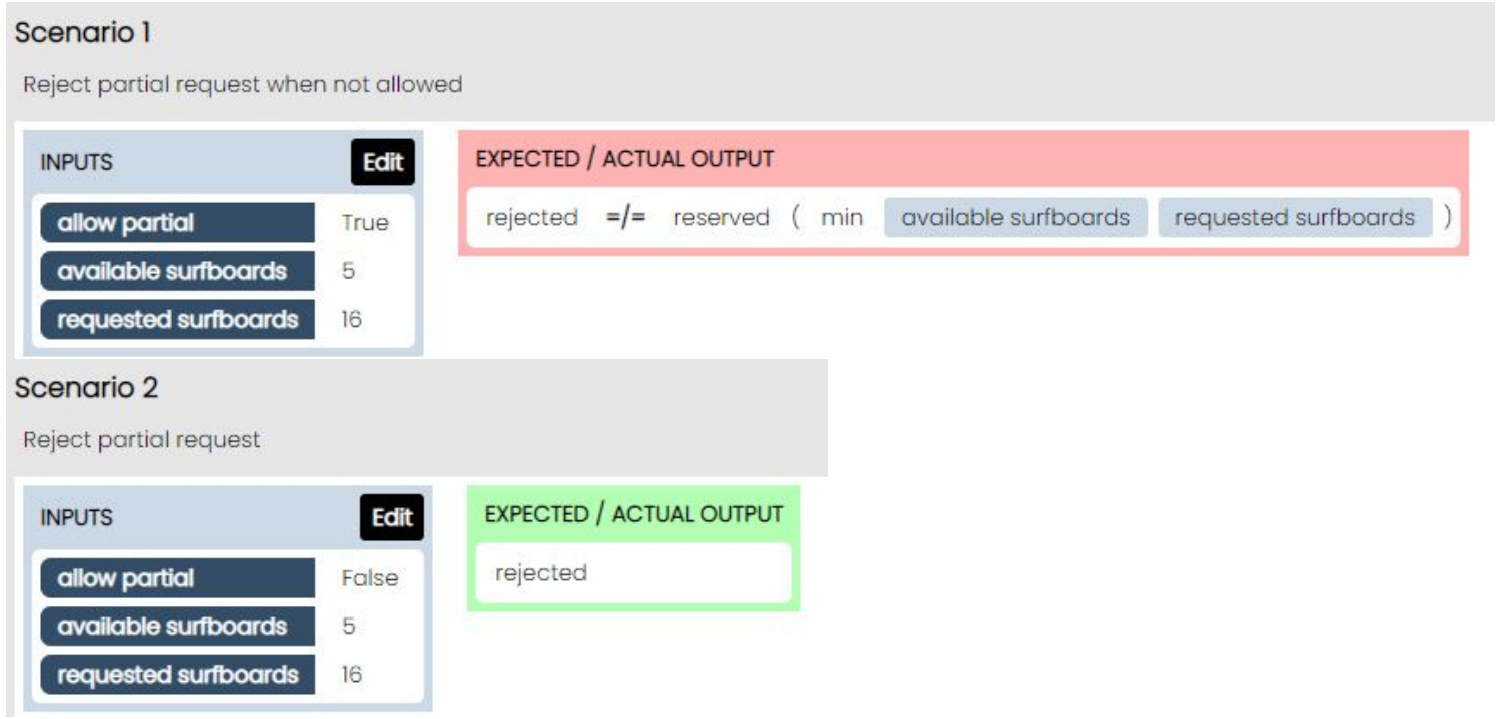
Verifying the logic

- When the spec looks right it's time to verify
- FP has built-in guarantees
- Semi-automated tests

- BDD style checks in seconds
- Reuse visualization to display results

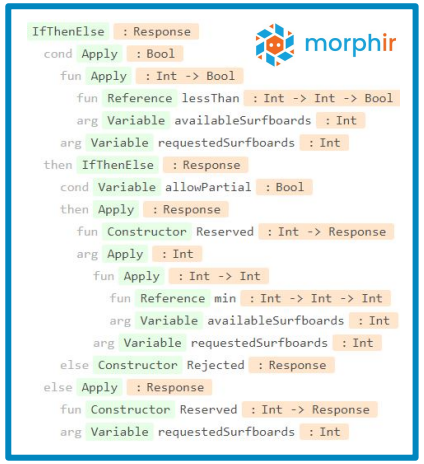


logic driven tests



Transpiling the logic

- Data is easy to translate back to logic
- Code generators automate an error-prone manual process
  - they are reliable, testable and scalable
- Allows you to change stack without risking business logic
- Tech PoCs can be real instead of mock-ups



code generation

```
var request = F3(
    function (allowPartial, availableSurfboards, requestedSurfboards) {
        return (_Utils_cmp(availableSurfboards, requestedSurfboards) < 0) ?
        (allowPartial ? Reserved(
            A2($elm$core$Basics$min, availableSurfboards, requestedSurfboards)) : Rej
            : Reserved(requestedSurfboards);
    });
```

JS

```
def request(
  allowPartial: morphir.sdk.Basics.Bool
)(
  availableSurfboards: morphir.sdk.Basics.Int
)(
  requestedSurfboards: morphir.sdk.Basics.Int
): morphir.reference.model.presentations.LondonFintechMeetup.Response =
  if (morphir.sdk.Basics.lessThan(availableSurfboards)(requestedSurfboards)) {
    if (allowPartial) {
      morphir.reference.model.presentations.LondonFintechMeetup.Reserved(morphir.sdk.Basics.min(availableSurfboards)(requestedSurfboards))
    } else {
      morphir.reference.model.presentations.LondonFintechMeetup.Rejected
    }
  } else {
    morphir.reference.model.presentations.LondonFintechMeetup.Reserved(requestedSurfboards)
  }
```

Scala

FINOS

Fintech
Open Source
Foundation

Get Involved

- FINOS Open Reg Tech
- Application modeling
- New language and platforms
- More interesting tools

Landing page

`http://morphir.finos.org`

GitHub

`https://github.com/finos/morphir`