

Reactive GUI application architecture with dataflow constraints

Marcel Weiher

@mpweiher

Reactive GUI application architecture with dataflow constraints

ui = | = memory-model.

memory-model ::= persistence.
persistence | = memory-model.

backend = | = memory-model.

Thank you!

Reactive Dataflow Constraints

- Reactive Programming
- Dataflow
- Dataflow Constraints

Reactive Dataflow Constraints

- **Reactive Programming**
- Dataflow
- Dataflow Constraints

Reactive Programming

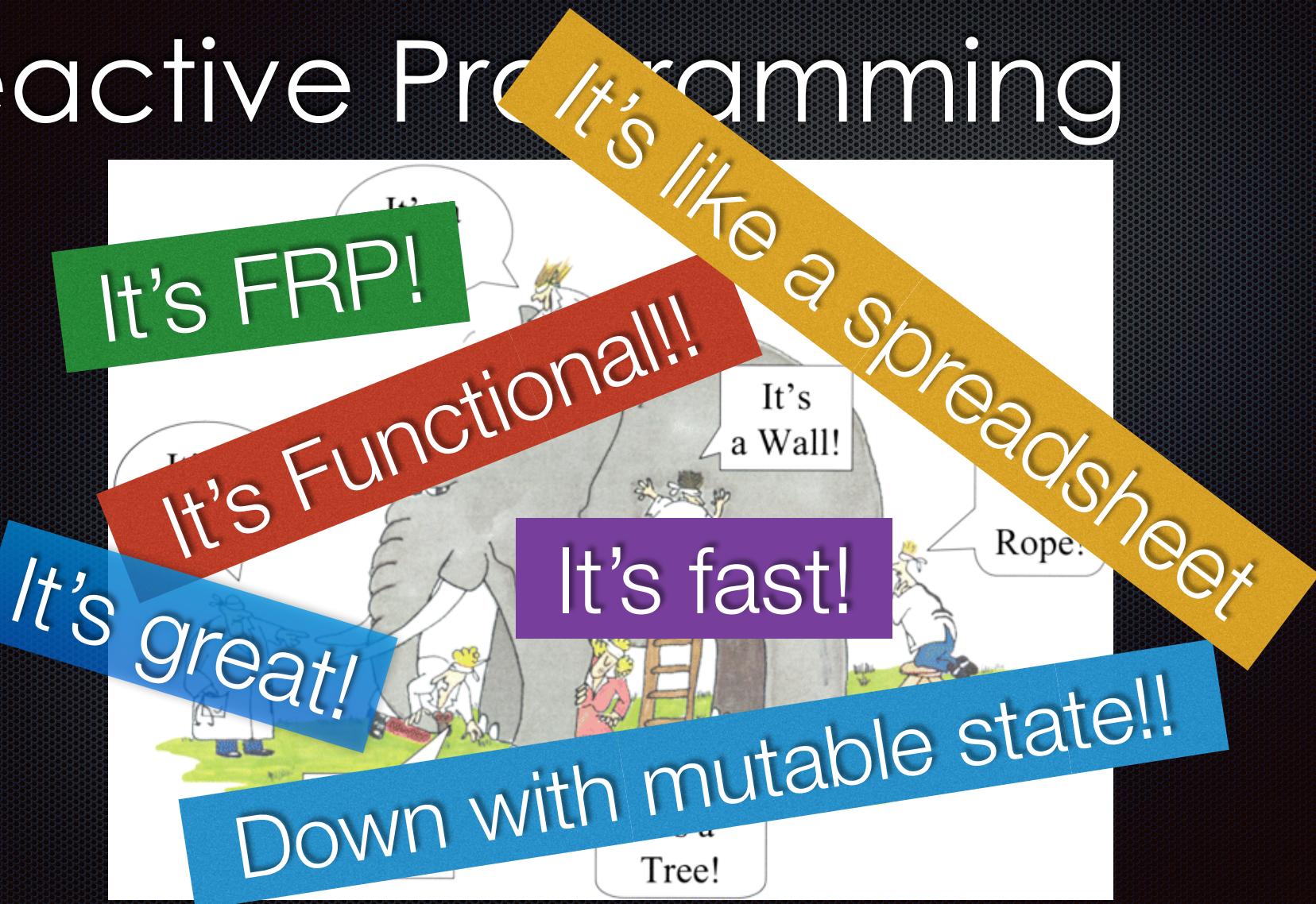
It's FRP!

It's Functional!!

It's great!

It's fast!

Down with mutable state!!



Reactive Programming

- *FRP was somewhat of a misnomer, better: “Functional Temporal Programming”*
 - Conal Elliot, creator of FRP
- *Rx is not functional and not reactive*
 - Erik Meijer, creator of Rx
- *RAC, Elm, Rx etc. have nothing to do with FRP*
 - Conal Elliot and Erik Meijer

Reactive Programming



Reactive Programming

- Reactive is a system characteristic: Harel
- “Reactive” style is a variant of synchronous dataflow
Esterel (imperative, Airbus), Lustre (functional), Ptolemy (dsp)
- Rx, etc. are not synchronous in that way

Reactive Dataflow Constraints

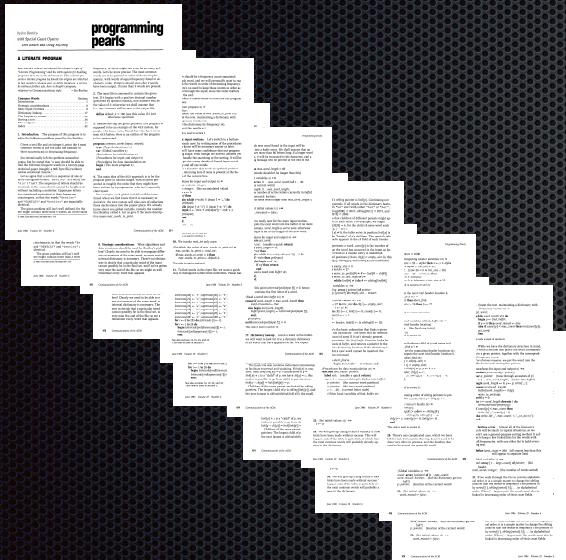
- Reactive Programming
- **Dataflow**
- Dataflow Constraints

Dataflow

- Unix Pipes and Filters
- Wunderlist

Dataflow Unix Pipes and Filters

Bentley's Challenge

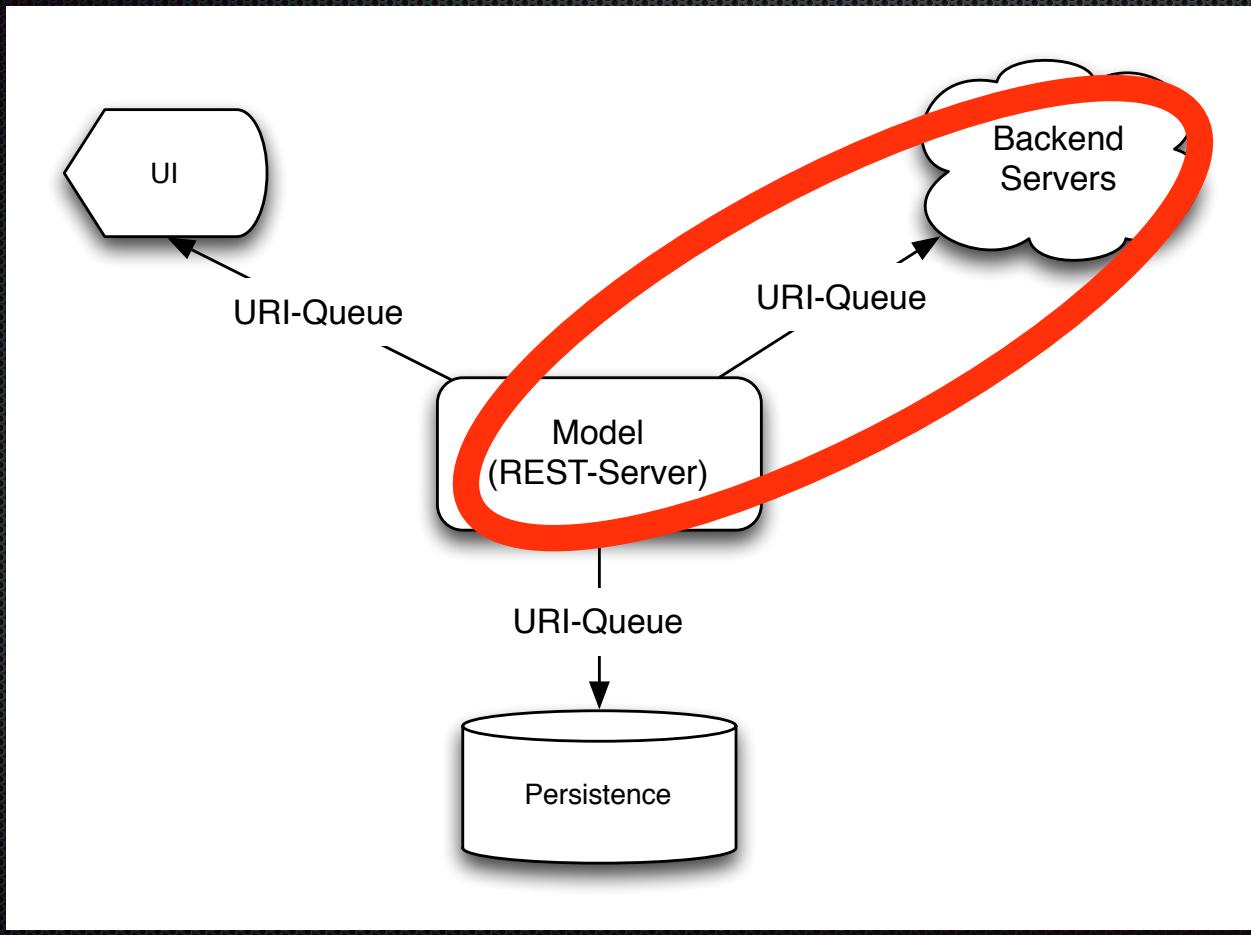


```
tr -cs A-Za-z '\n' |  
tr A-Z a-z |  
sort |  
uniq -c |  
sort -rn |  
sed ${1}q
```

Don Knuth

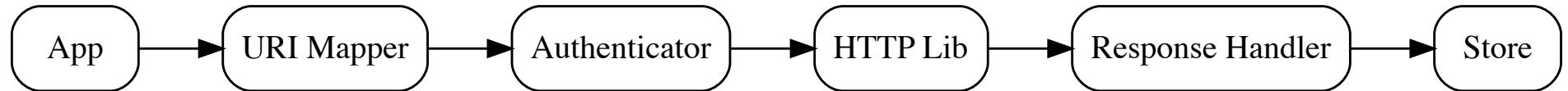
Doug McIlroy

Wunderlist

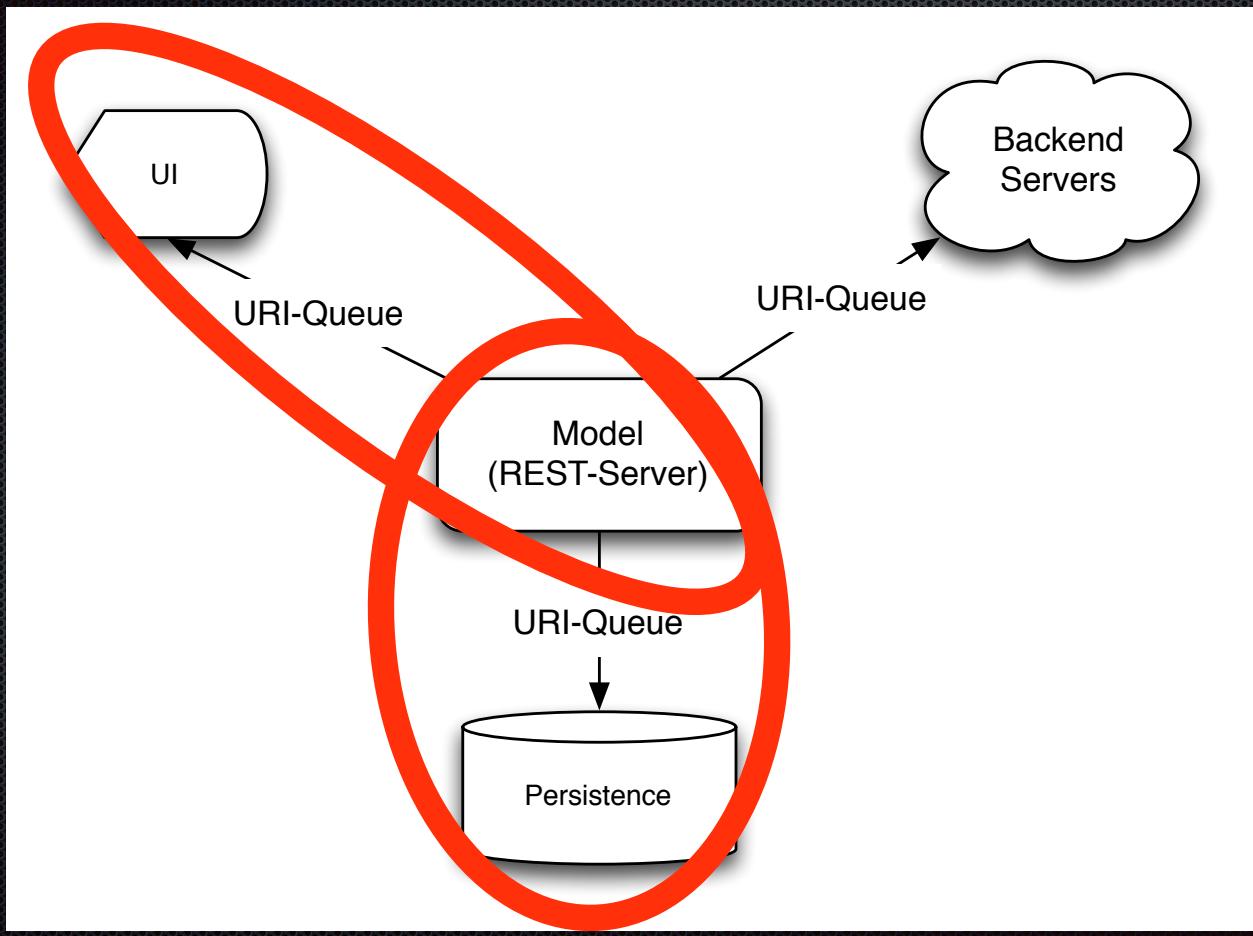


Dataflow

In-process



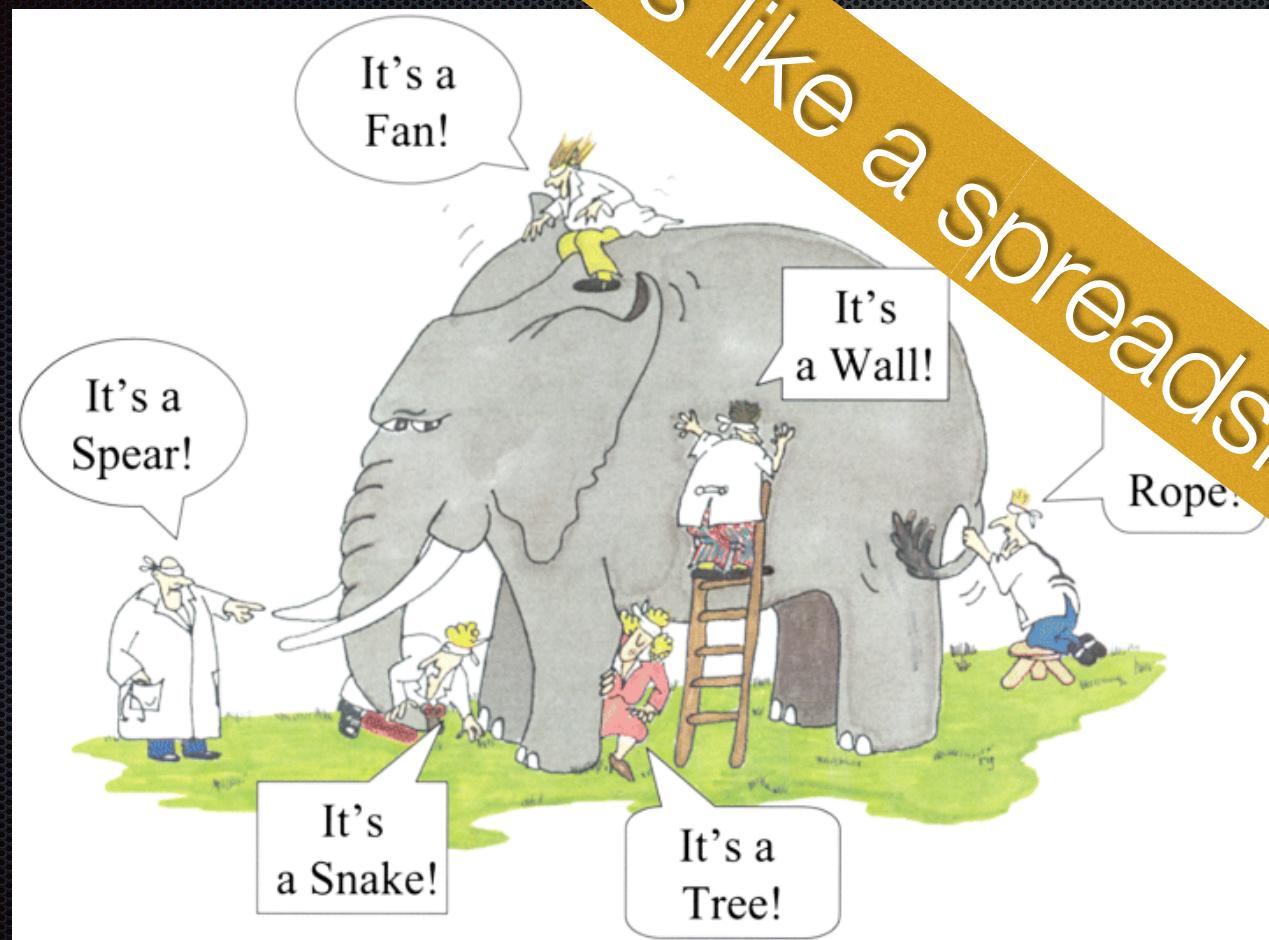
Wunderlist



Reactive Dataflow Constraints

- Reactive Programming
- Dataflow
- **Dataflow Constraints**

Dataflow Constraints



Dataflow Constraints

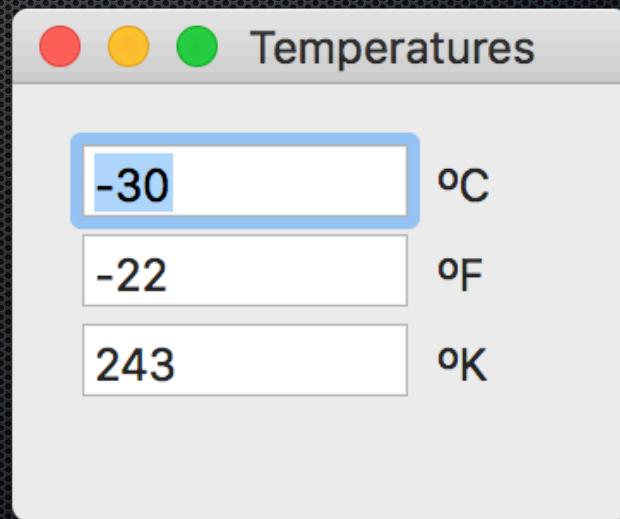
ui = | = memory-model.

memory-model ::= persistence.
persistence | = memory-model.

backend = | = memory-model.

Architectural Mismatch Example

- Model
- Add UI
- Add persistence
- Extend model



Temperature Converter Model

```
1  - f:degreesF {
2      self c:(degreesF - 32) / 1.8.
3  }
4  - f {
5      ^ self c * 1.8 + 32.
6  }
7  - c:degreesC {
8      ivar:c := degreesC.
9  }
10 - c {
11     ^ivar:c.
12 }
```

Objective-Smalltalk

- Smalltalk dialect for architectural experimentation
keyword messages, ^ = return,
- Define Connectors
| = = | = ...
- Polymorphic Identifiers (URLs as identifiers)
`http://www.modularity.info`
`ivar:c`
`defaults:celsius`

Temperature Converter

Add UI

```
- changedF:sender {
    self f:sender intValue.
}

- changedC:sender {
    self c:sender intValue.
}

- f:degreesF {
    self c:(degreesF - 32) / 1.8.
    ivar:ui/celsiusTextField/intValue := self c.
}

- f {
    ^ self c * 1.8 + 32.
}

- c:newValue {
    ivar:c := newValue.
    ivar:ui/fahrenheitTextField/intValue := self f.
}

- c {
    ^ ivar:c.
}
```

Temperature Converter

Minimize UI updates

```
- f:degreesF {
    self basicC:(degreesF - 32) / 1.8.
    ivar:ui/celsiusTextField/intValue := self c.
}
- c:newValue {
    self basicC:newValue.
    ivar:ui/fahrenheitTextField/intValue := self f.
}
- basicC:newValue {
    ivar:c := newValue.
}
```

Temperature Converter

Add Persistence

```
- basicC(newValue {
    ivar:c := newValue.
    defaults:c := ivar:c.
    self updateUI.
})
```

Temperature Converter

Add Kelvin (1)

```
- changedF:sender {
    self f:sender intValue.
}
- changedC:sender {
    self c:sender intValue.
}
- changedK:sender {
    self k:sender intValue.
}
```

Temperature Converter

Add Kelvin (2)

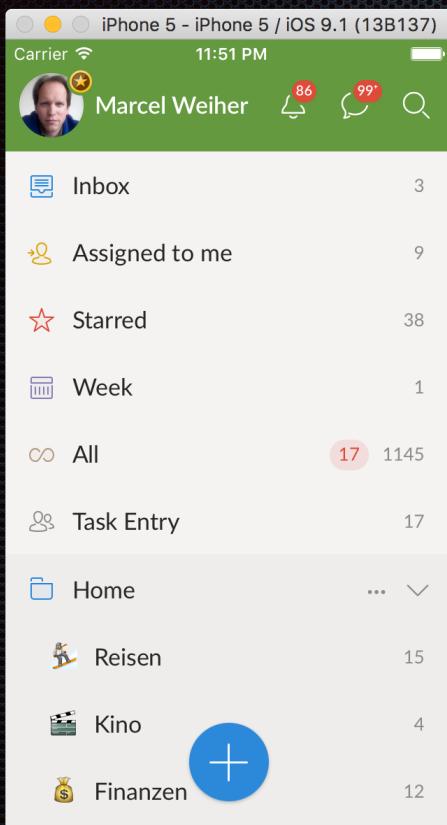
```
- f:degreesF {
    self basicC:(degreesF - 32) / 1.8.
    ivar:ui/celsiusTextField/intValue := self c.
    ivar:ui/kelvinTextField/intValue := self k.
}
- f {
    ^ self c * 1.8 + 32.
}
- k:degreesK {
    self basicC:(degreesF - 273.15.
    ivar:ui/celsiusTextField/intValue := self c.
    ivar:ui/fahrenheitTextField/intValue := self f.
}
- k {
    ^ self c + 273.15.
}
- c:newValue {
    self basicC:newValue.
    ivar:ui/fahrenheitTextField/intValue := self f.
    ivar:ui/kelvinTextField/intValue := self k.
}
```

WTF?

Architectural Mismatch in Interactive Programs

- Programs = Data + Algorithms + Architecture (Chatty, EIS '07)
- Architectural style of interactive systems
 - ≠
Style of programming language(s)
- “Tyranny of the call/return architectural style”

Wunderlist Client Architecture



$$\stackrel{?}{=} y=f(x)$$

Constraints

- Specify Relationships to Maintain (Declarative)
- Dataflow Constraints = Spreadsheets
- Build systems / make

Temperature Converter Model Constraints

```
celsius    |= fahrenheit * 5/9 - 32.  
fahrenheit |= (celsius + 32) * 9/5.
```

Temperature Converter

Add UI

```
celsiusTextField/intValue    =|= celsius.  
fahrenheitTextField/intValue =|= fahrenheit.
```

```
celsius    |= fahrenheit * 5/9 - 32.  
fahrenheit |= (celsius + 32) * 9/5.
```

Temperature Converter

Add Persistence

```
celsius          := defaults:celsius.  
defaults:celsius |= celsius.
```

```
celsiusTextField/intValue    =|= celsius.  
fahrenheitTextField/intValue =|= fahrenheit.
```

```
celsius      |= fahrenheit * 5/9 - 32.  
fahrenheit   |= (celsius + 32) * 9/5.
```

Temperature Converter

Add Kelvin

```
celsius          := defaults:celsius.  
defaults:celsius |= celsius.  
  
celsiusTextField/intValue   =|= celsius.  
fahrenheitTextField/intValue =|= fahrenheit.  
kelvinTextField/intValue   =|= kelvin.  
  
celsius    |= fahrenheit * 5/9 - 32.  
fahrenheit |= (celsius + 32) * 9/5.  
kelvin     |= celsius - 273.  
celsius    |= kelvin + 273.
```

Temperature Converter Group

```
memory-model      := persistence.  
persistence       |= memory-model.  
  
celsiusTextField/intValue   =|= celsius.  
fahrenheitTextField/intValue =|= fahrenheit.  
kelvinTextField/intValue    =|= kelvin.  
  
celsius      |= fahrenheit * 5/9 - 32.  
fahrenheit    |= (celsius + 32) * 9/5.  
kelvin        |= celsius - 273.  
celsius      |= kelvin + 273.
```

Temperature Converter Group

```
memory-model      := persistence.  
persistence       |= memory-model.  
  
ui                = |= memory-model.  
  
celsius          |= fahrenheit * 5/9 - 32.  
fahrenheit       |= (celsius + 32) * 9/5.  
kelvin           |= celsius - 273.  
celsius          |= kelvin + 273.
```

Temperature Converter Group

memory-model := persistence.
persistence |= memory-model.

ui = |= memory-model.

memory-model = |= "consistency"

Wunderlist Architecture

memory-model := persistence.
persistence |= memory-model.

ui = |= memory-model.

memory-model = |= backend

Constraint Composition

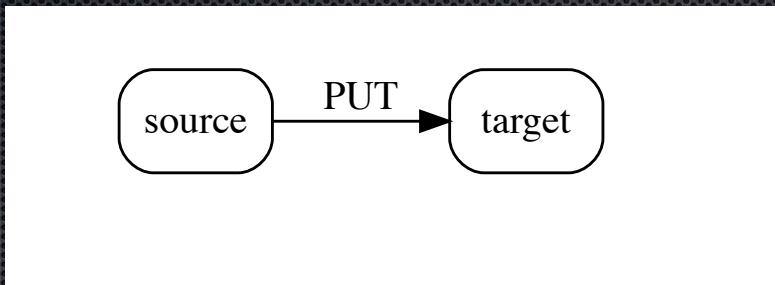
- Separate constraint interface and implementation
- Implement constraints between composite objects as constraints between (some of) their components

Dataflow Constraints in Wunderlist

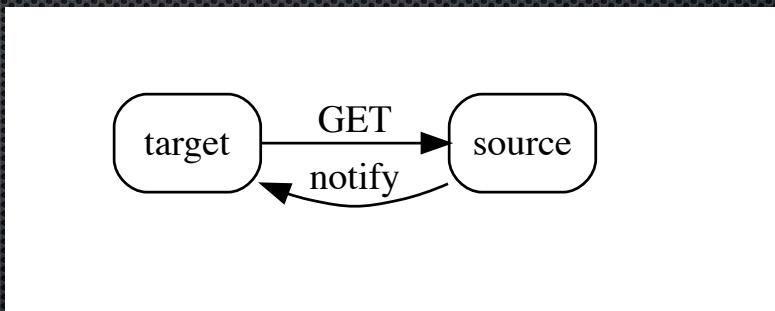
- No linguistic or solver support
- Push vs. pull constraints
- In-Process REST
- URI queues for notifications

Dataflow Constraints in Wunderlist

- Push



- Pull

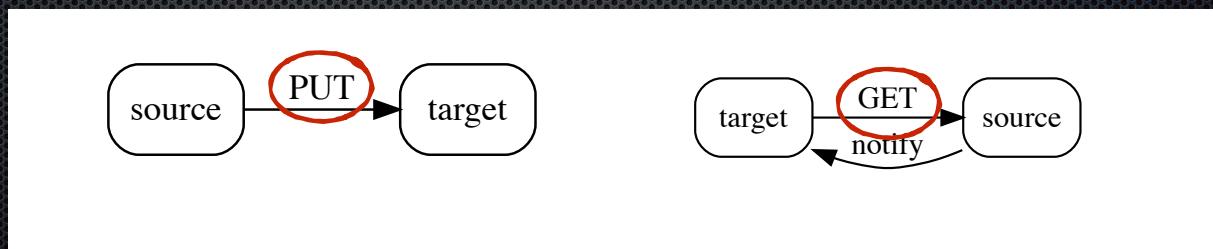


In-Process REST

- Use URIs (StoreReference) to refer to model
- URIs are structured:

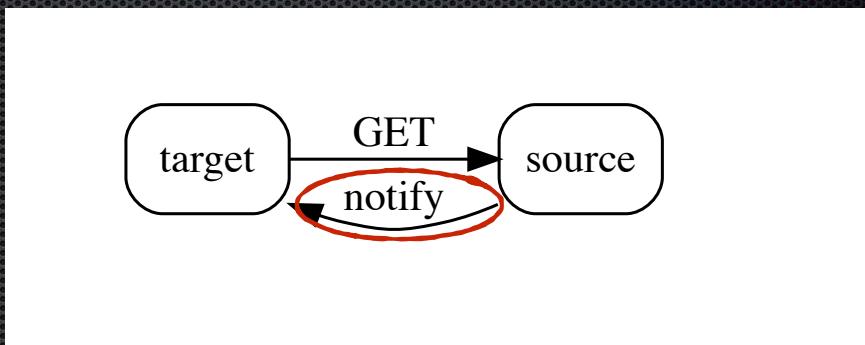
wlstore:<entity>/container/<id>/object/<id>

- URIs can refer to single objects or groups

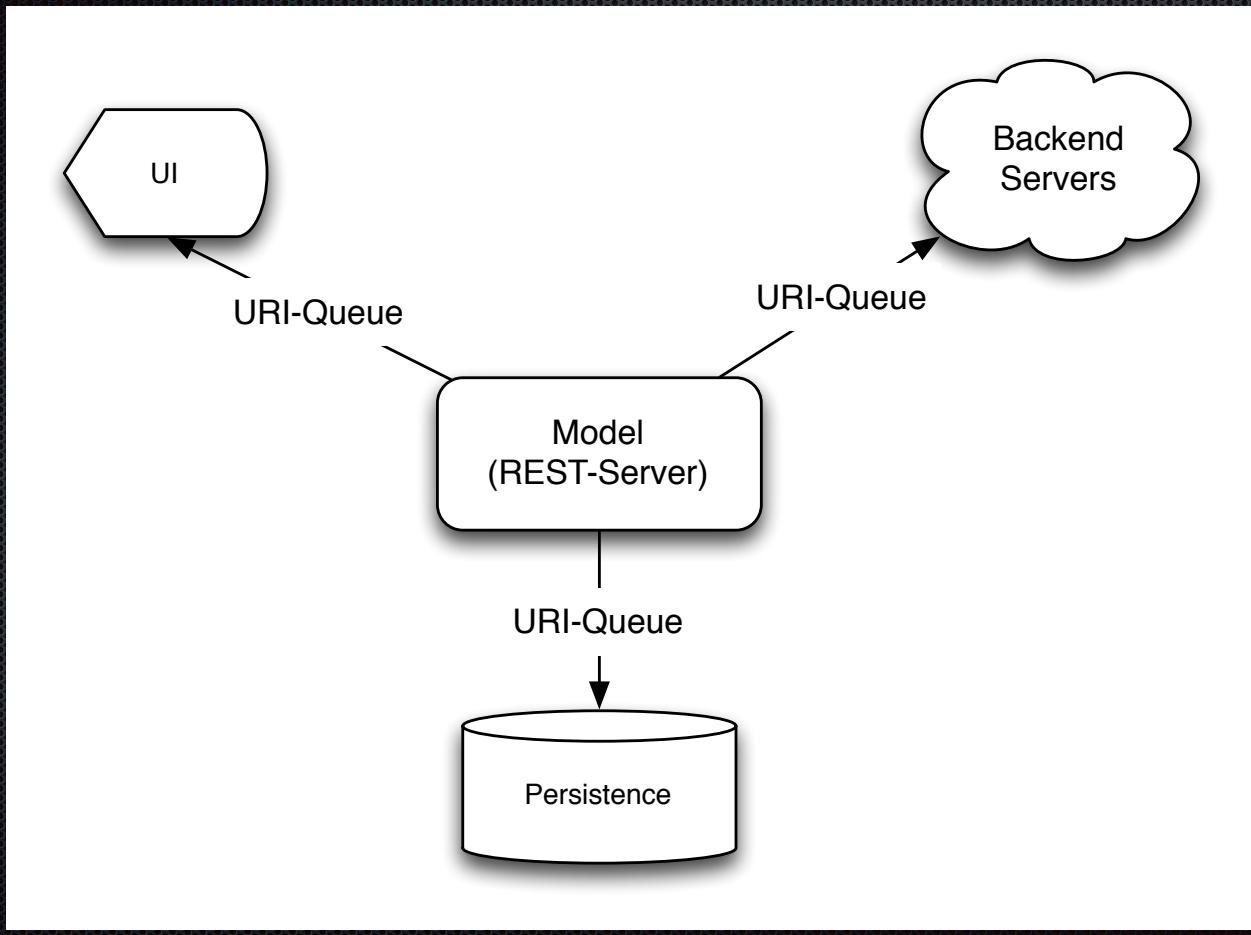


URI Queues

- Asynchronous
- Persistable
- Uniqued
- “Bucketized”
- Dynamic coalescing/throttling (UI)



Wunderlist



Architecture Dataflow Constraints

```
memory-model      := persistence.  
persistence       | = memory-model.  
ui                |=| = memory-model.  
memory-model      |=| = backend
```

- <http://objective.st/>

@mpweiher