Expressive Systems

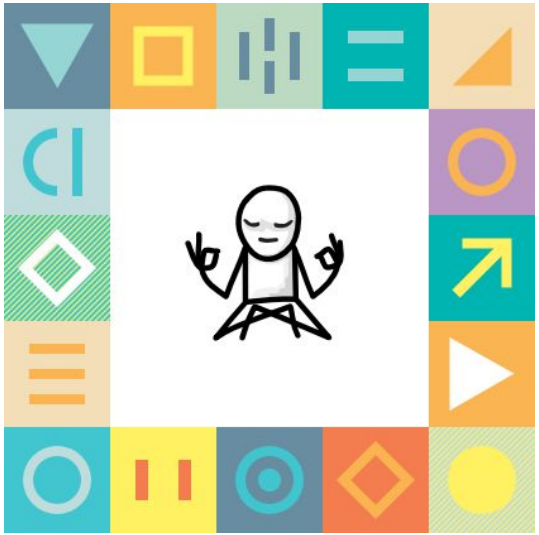
Rapid Development of Flexible Business Applications

Pharo Days 2022

Erik Stel / Jonathan van Alteren

Agenda

- Object Guild
- Expressive Systems characteristics
- Architecture overview
- Hands-on demo
- Questions & answers



Object Guild

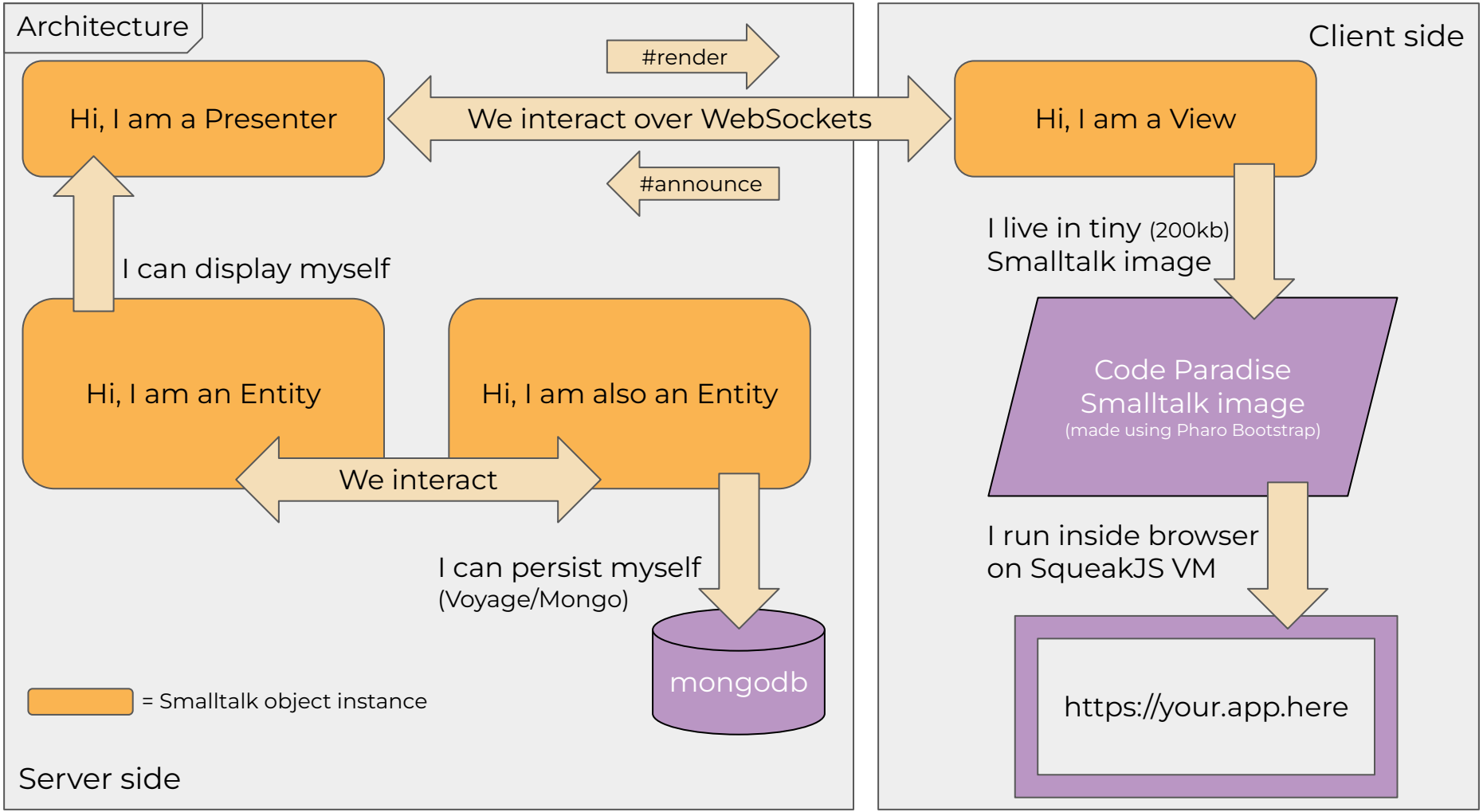
OBJECT GUILD

- Purpose-driven
- Human centered
  - Psychology and anthropology
- Inspired by...
  - David West (Object Thinking)
  - Richard Pawson (Naked Objects)
  - Rebecca Wirfs-Brock (Designing Object-Oriented Software)

Expressive Systems characteristics

- Focus on business domain model
- Auto-generated web UI
- Direct manipulation of objects
- Transparent persistence
- Security built-in by design

CAUTION! Under active development :-)


Architecture
Client side
#render
Hi, I am a Presenter
We interact over WebSockets
Hi, I am a View
#announce
I can display myself
I live in tiny (200kb)
Smalltalk image
Hi, I am an Entity
Hi, I am also an Entity
Code Paradise
Smalltalk image
(made using Pharo Bootstrap)
We interact
I can persist myself
(Voyage/Mongo)
I run inside browser
on SqueakJS VM
mongodb
= Smalltalk object instance
https://your.app.here
Server side

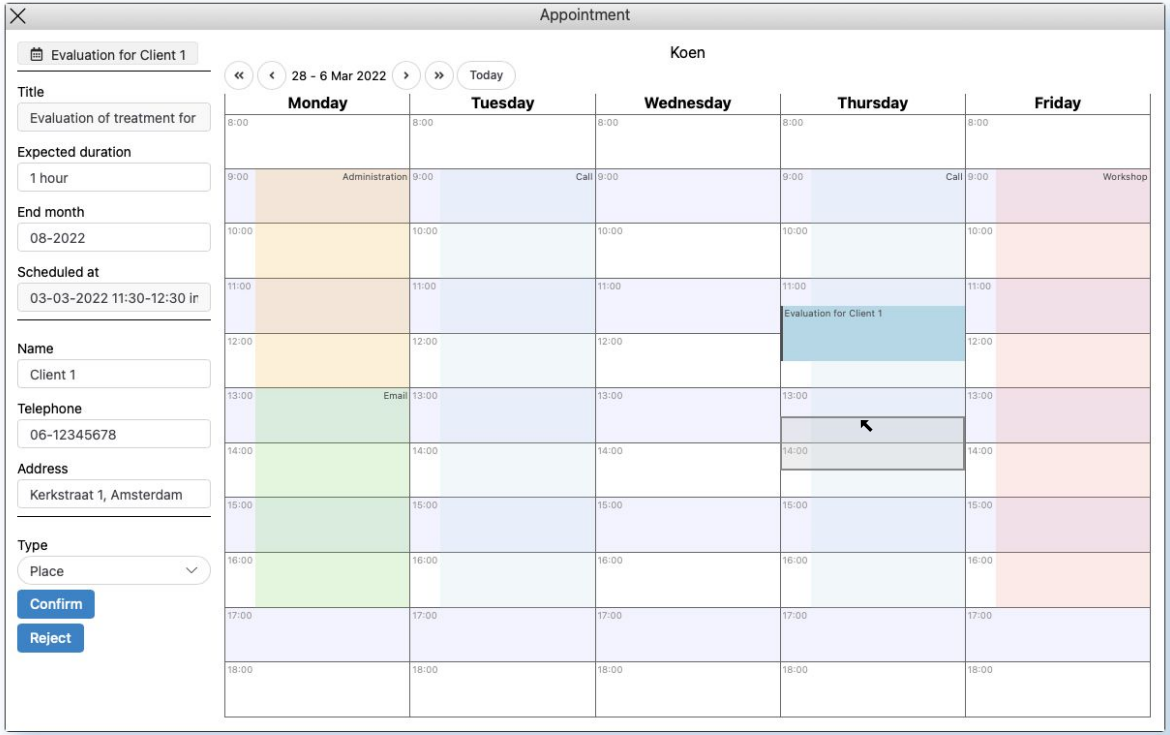
Travel Booking domain - Happy Trips

- Based on Richard Pawson's demo of Naked Objects
- Domain entities
  - Customer
  - Booking
  - City
  - Passenger
  - (Payment)

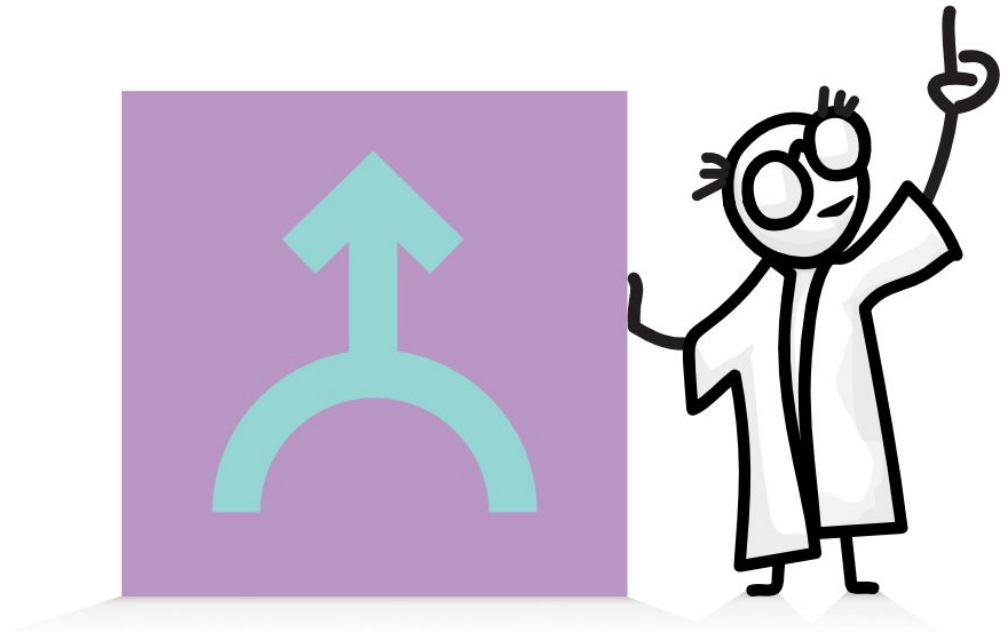
Hands-on demo - transcript

- Showing around
  - Entity classes collection
  - Entity instance collection
    - Sort
    - Paging
  - Entity form
    - Flexible input
    - Human friendly errors
  - New entity (click / drag)
  - Associate entity (drag)
  - Actions
  - Custom presenter (screenshot)
- Hands-on part
  - Add Payment entity
  - Extend with...
    - Element
      - Amount
      - Status
    - Action
      - Pay

Hands-on demo - Screenshot custom presenter



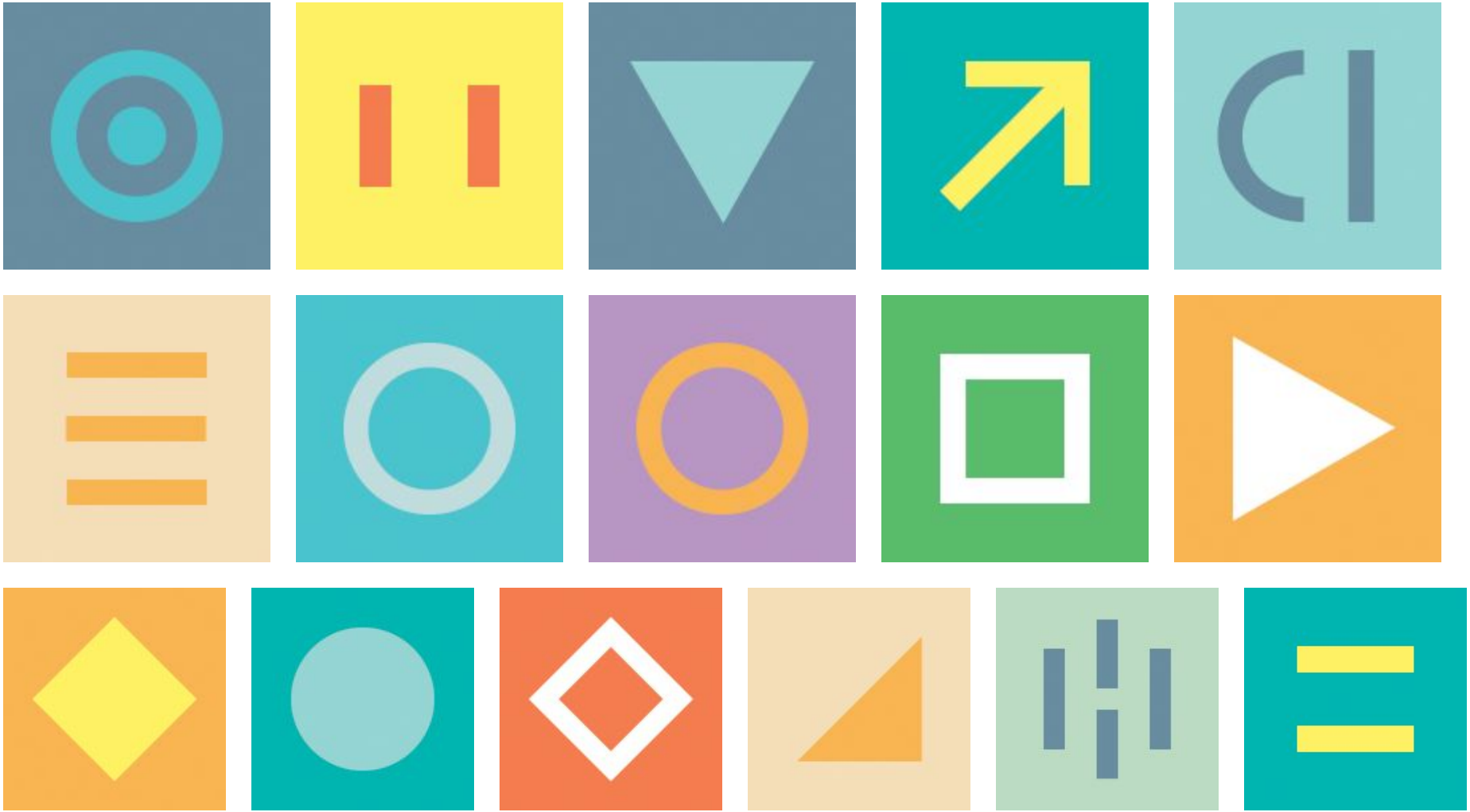
Questions?



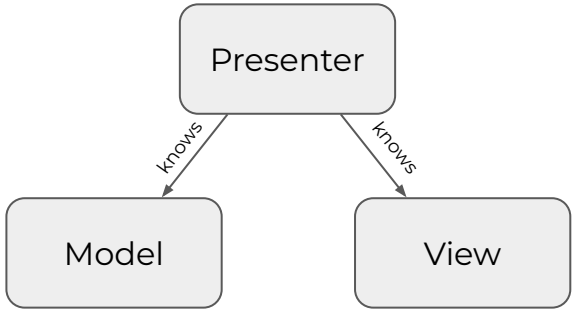
OBJECT GUILD

Do you want to know more? Contact us at: info@objectguild.com

Or go to: https://objectguild.com



Model View Presenter explained

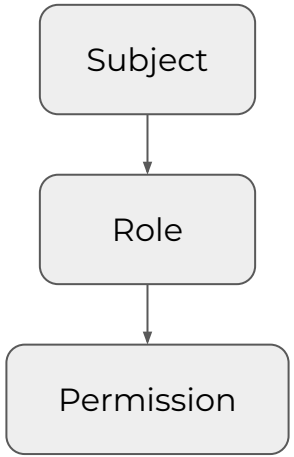


- MVP model
  - Model announces updates
  - Presenter listens to Model and renders/listens to View
  - View announces user interaction
- Views are defined using HTML and CSS
  - Updates to HTML and CSS also take effect directly
- Views have there logic defined in Smalltalk code
  - No Javascript is needed (is part of Code Paradise)
- Views are only the ‘glass’
  - The Views are very thin (only Strings are rendered) and have no knowledge about Entities.
- Updates in code take effect directly
  - When a Presenter renders its View any necessary classes and methods will be installed in the tiny Smalltalk image. There is no compiler needed in the client because View code is developed in the server side Pharo image. So only bytecodes are sent to the client. The SqueakJS VM is bytecode compatible with Pharo 8 (and work is being done on Pharo 9/10).

Security built-in by design

- Role-Based Access Control model
  - Subject has Role(s) and Principal(s)
  - Role has Permission(s)
  - User is a Subject and has Credential(s)
- Permission checked at the method level
  - Booking >> #confirm

```
self assertPermission: #confirm.
status := status toConfirmed.
customer sendConfirmationFor: self
```



Abstract

We are developing the Expressive Systems framework, inspired by Richard Pawson's work on Naked Objects. It allows us to rapidly develop flexible applications by focusing on the design of behaviorally complete objects in the business domain. By using a novel web application architecture, it allows direct manipulation of business objects by the user.