October 2nd, 2015

Jiri Jetmar, CEO Smartnerds GmbH
Paul Merlin, Title/Role here

Agenda

- **What is Apache Zest ?**
- Zest Libraries & App Building Strategies
- Sample App – Live Coding Session
- Zest Community, Real World Apps & Outlook

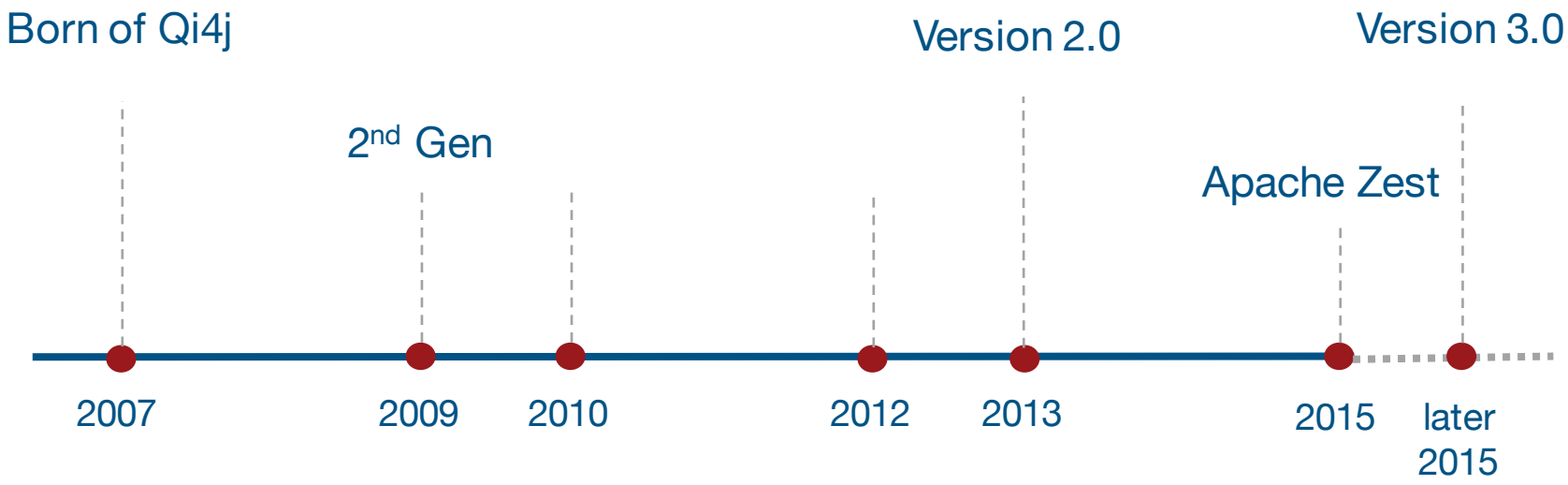
"Genius is one percent inspiration and ninety–nine percent perspiration."

Thomas A. Edison

Apache Zest™ Fact Sheet

:: History

- Apache Zest was born as **Qi4j** in 2007
- Founded by **Richard Öberg** (EJBoss, later JBoss) and **Niclas Hedhman**
- Version 1.0, 1.1 and 1.2 in 2010
- Version 2.0 in 2013
- Since beginning of 2015 a “Gold” Apache Foundation Project // JJ
- Current Version 2.1, 3.0 in 2015 with lots of new features



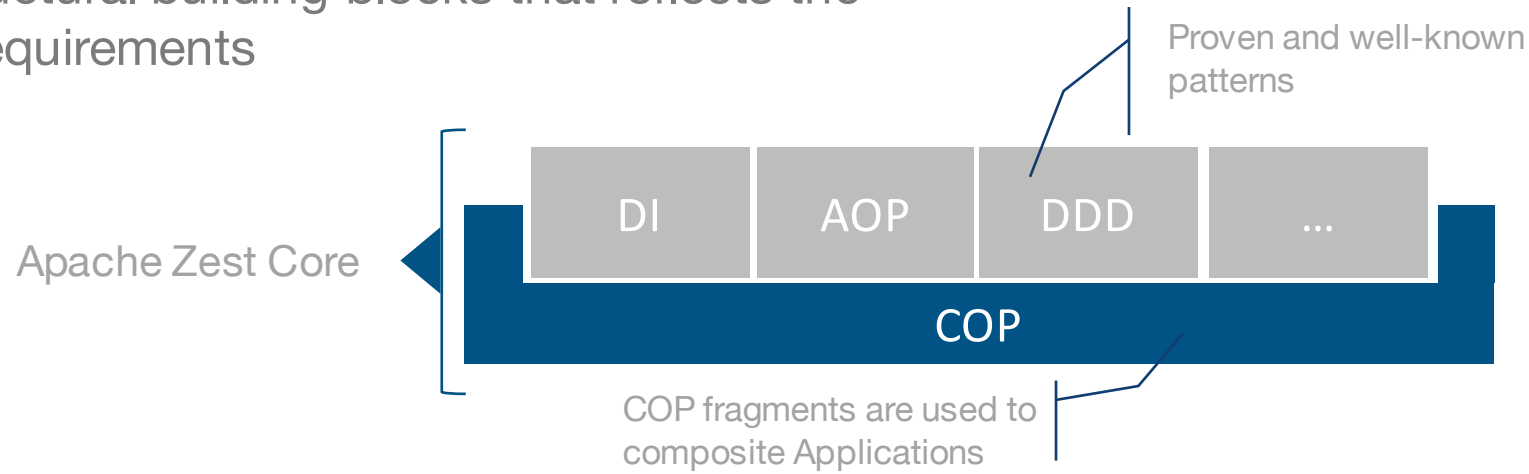
Apache Zest™ – It is about solving (business) problems

:: (Re-) Usage of proven patterns

- Zest is nothing new. It is an logical next step based on existing and proved patterns and ideas
- Zest is an Java implementation of the **Composite Oriented Programming** (COP) pattern
- COP is a programming model that allows the creation of rich domain models

:: Business problem centric approach

1) Start with the business problem
2) Use the terminology from Domain Driven Design (DDD)
3) Allow developers to implement the Domain Model directly in code using that Terminology
4) Plug infrastructural building blocks that reflects the needs and requirements



Apache Zest™ – What sort of problems to solve with Zest ?

:: Rich Domain Models

- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diamZest is an
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam

:: High ratio of DM changes over time

- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diamZest is an
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam

:: Transactional, integrative and API centric Applications

- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diamZest is an
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
- Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam

Apache Zest™ & COP Overview

:: COP in one sentence

"Composite Oriented Programming allows developers to work with 'fragments', smaller than classes, and 'compose' fragments into larger 'composites' which acts like the regular objects."

Niclas Hedhman

:: COP & Apache Zest

- The most basic element in Zest is the Composite
- A Composite is created by composing a number of Fragments
- Mixings are Fragments that can handle method invocations
- Modifiers are Fragments that modify method invocations (Decorator pattern) – Constraints, Concern, SideEffects

Apache Zest™ Basic Fragments

:: Fragments

- **@Constraint** validates method arguments
  - Can have many Constraints per argument
  - Uses annotations to trigger
  - Cooperate with concern for failure actions
- **@Concerns** intercept method calls
  - Allowed to modify arguments and i.) return values, ii.) return with calling in chain, iii.) throws exceptions
- **@SideEffects** are called after a method call has finished
  - Cannot change method arguments or return value
  - Cannot throws exceptions
  - Can inspect exceptions and return values
  - May be asynchronous
- **@Mixins** implements Composite interfaces
  - A Mixin may implement one interface, many interfaces or just some methods
  - May contain Composite state, such as Property and Association instances
  - May be Composite private – not exposed in Composite interface

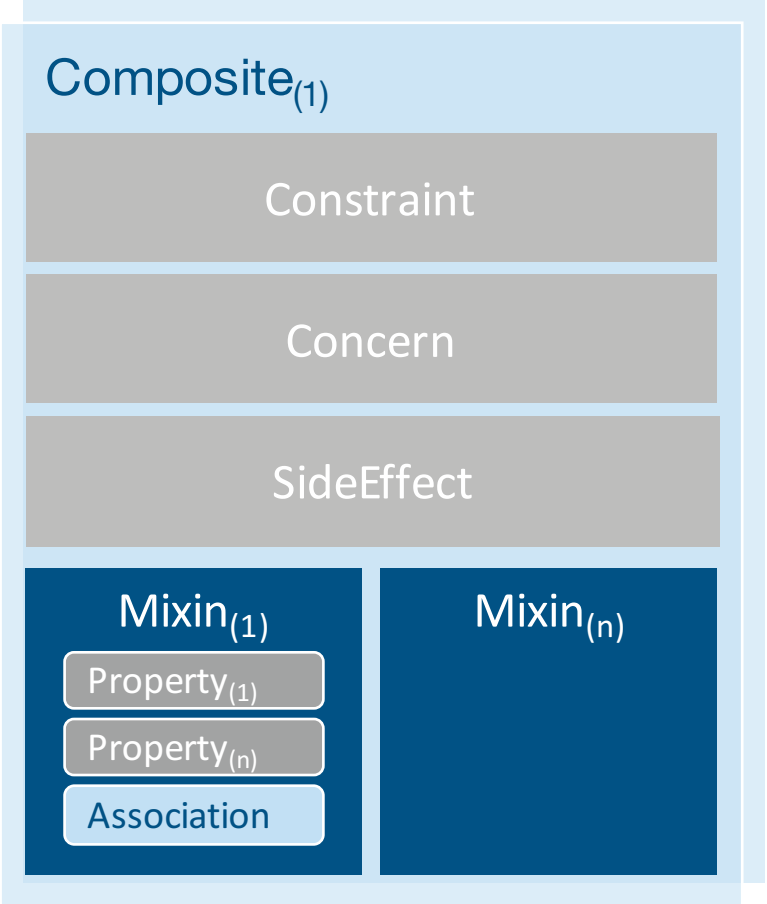


Figure XX : Composite Fragments

Apache Zest™ Structure Composition

:: Application Structure

- Composites define the internals of objects
  - Explicit Composite meta types : Value, Entity, Service and Transient
- Composites resides in Modules
- Modules can be grouped into Layers
  - One or more Layers per Application
  - Zero, one or more Modules per Layer
- Visibility and dependency of Composites between structures is controlled
  - Declaration of Visibility of Composites

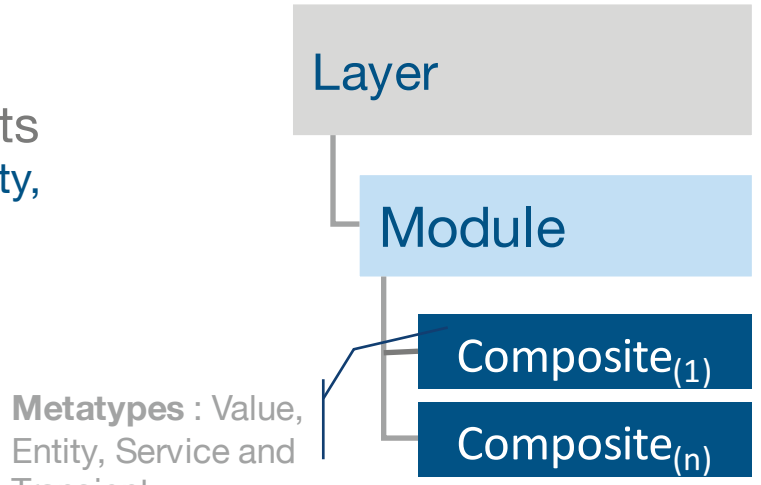


Figure XX : Structure Composition

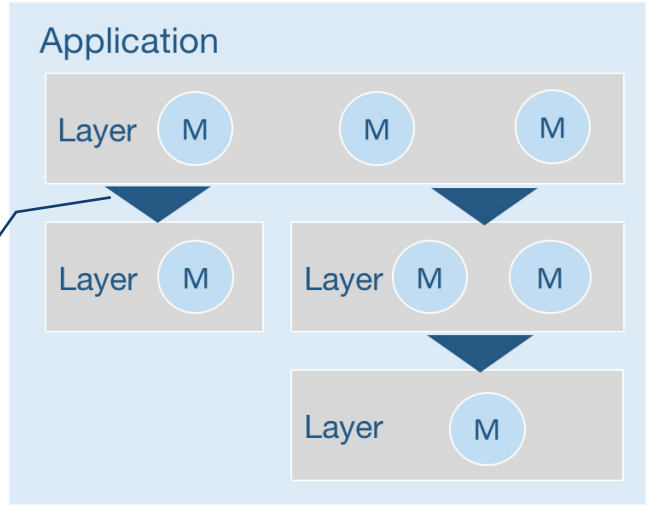


Figure XX : Application Structering

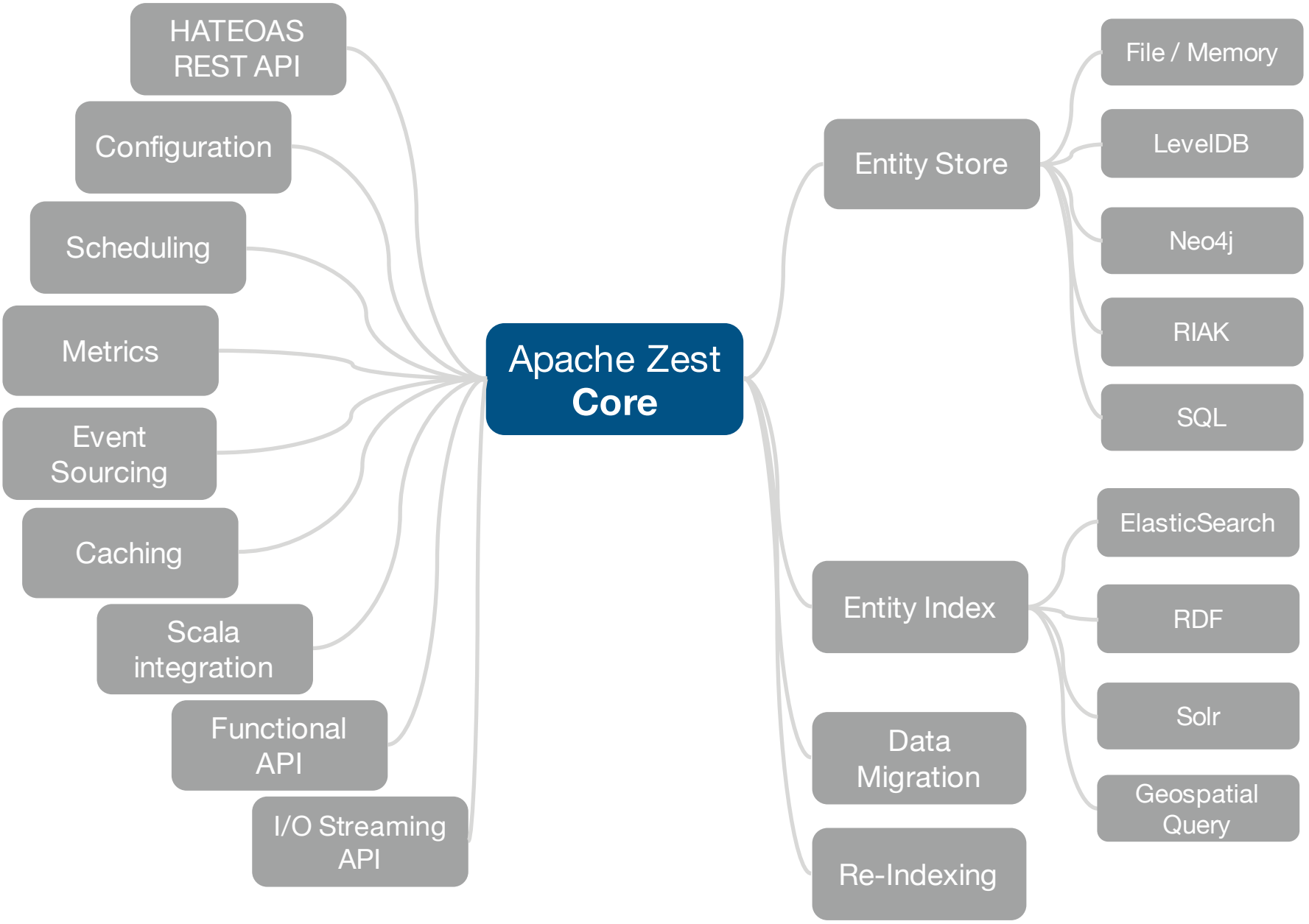
Agenda

- What is Apache Zest ?
- **Apache Zest Libraries & App Building Strategies**
- Sample App – Live Coding Session
- Zest Community, Real World Apps & Outlook

"Genius is one percent inspiration and ninety–nine percent perspiration."

Thomas A. Edison

Apache Zest™ Library ecosystem (.. there is much more !)



Application Building – A bit „grey“ Theory

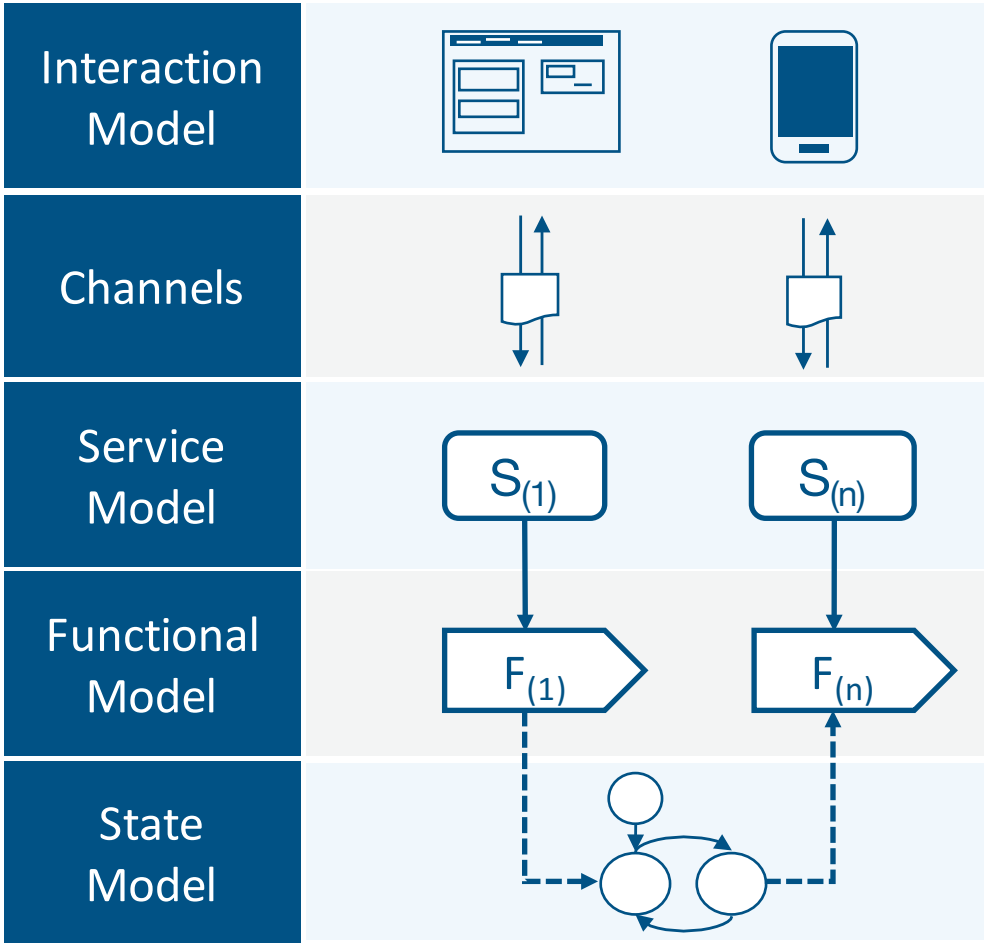


Figure XX : Top-Down Application Model

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore

Todo..

Application Building – ... How to do using Zest !

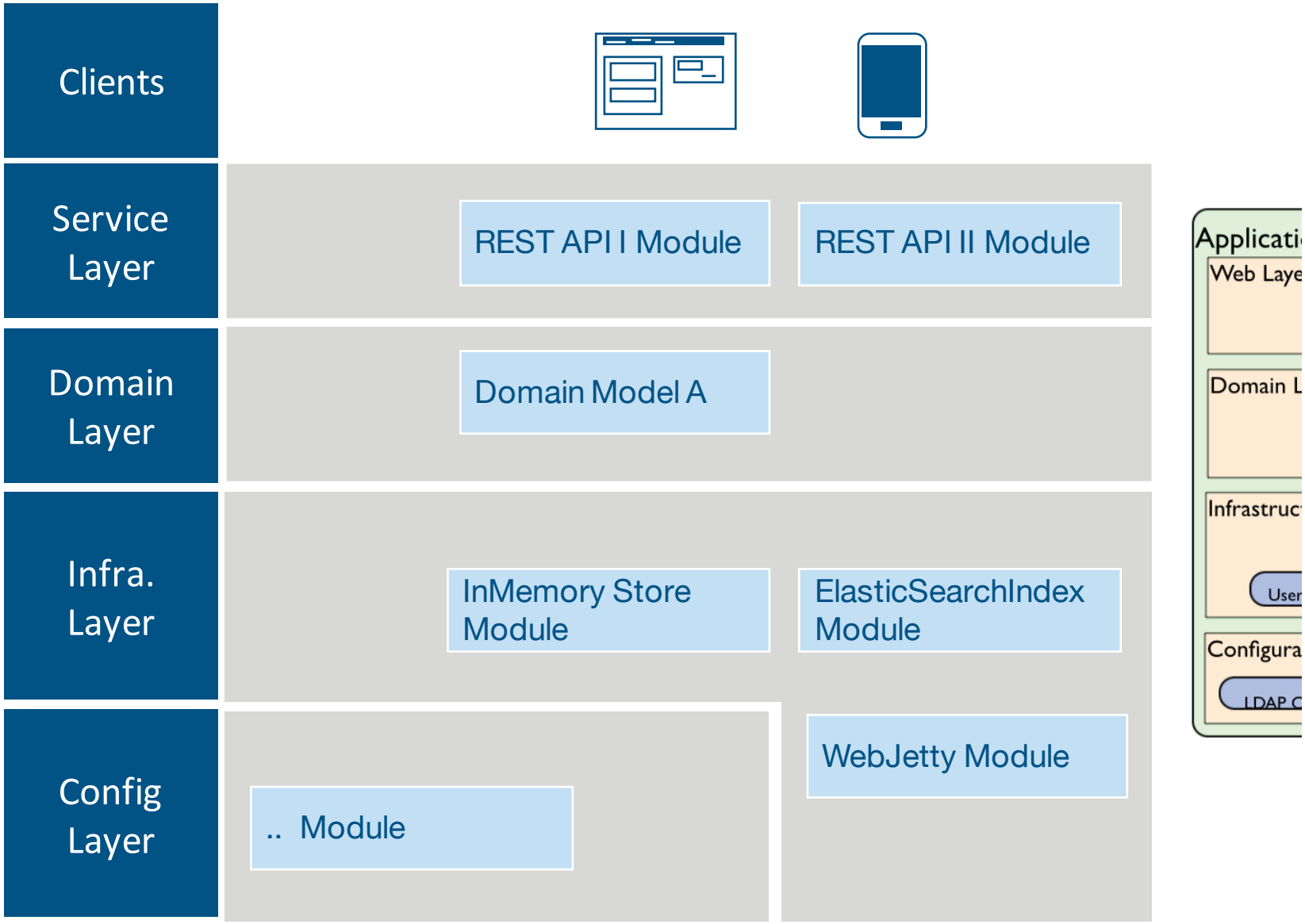


Figure XX : Concrete Zest-based Application structure

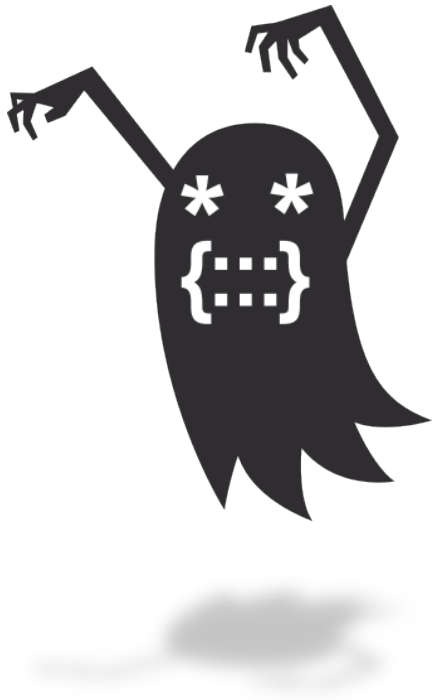
Agenda

- What is Apache Zest ?
- Zest Libraries & App Building Strategies
- **Sample App – Live Coding Session**
- Zest Community, Real World Apps & Outlook

"Genius is one percent inspiration and ninety–nine percent perspiration."

Thomas A. Edison

Sample App – Live Coding Session



$ git clone https://github.com/xxx/xxxx.git

Agenda

- What is Apache Zest ?
- Key Components, Building Blocks & Ecosystem
- Sample App – Live Coding Session
- **Zest Community, Real World Apps & Outlook**

"Genius is one percent inspiration and ninety–nine percent perspiration."

Thomas A. Edison

Apache Zest™ Community

:: Community

The current Zest developer community is rather small, but passionate individuals who are all convinced of Zest's merit and potential.

Over the years there has been **28** code contributors in total, of which **8** have been considered Core Developers, i.e. allowed to make changes on the Core runtime on trunk without hand-holding. No access control was in place to enforce that, and a **social contract worked very well**.

… so do not wait, JOIN us right now !!

We are looking forward to..

Who is using Apache Zest™ ?

Productive

http://smarpay.ch is a swiss company that offers mobile payment services in the vending business. Apache Zest is used to offer backoffice services like settlement and clearing for about 3k vending machines and about 350k mobile users.

In development

P* - http://dieparkuhr.de is a german Startup that offers services around the car parking business. Apache Zest is used for the entire backoffice with a large number of services like User-/Partner Management, Product Catalog, Reservation- & Scheduling System, Payment Services.

Apache Zest™ Outlook

:: Current Version is **2.1**

- First release of the Qi4j codebase under the ASF umbrella. Still uses org.qi4j.* for backward compability.
- A number of feature added (Complex Configuration Types, NamedAssiciations, toValue() and toEntity() conversions)
- Bugfixings and cleanups
- ..

:: Outlook for 3.0 (end of 2015)

- Full Java 8 Support (Lambda, Streaming API, ..)
- GeoSpatial Queries
- ...

Thank you !

See you at https://zest.apache.org !☺