

Beyond Ant & Maven: System Centric Java Development with Z2

Henning Blohm
ZFabrik Software KG
Submission #188

JAZOON'12



SET 2012

The z2-Environment is a radically new approach to solution life cycle management in Java

Motivation

- > Get rid of Build & Deploy
 - No out-of-sync Build infrastructure
 - No missing secondary assembly description
 - No incomplete deployments
- > Fix the Integration Leak:
 - Make sure you get the latest - always
 - Isolation as exception
- > Simplify life-cycle operations:
 - Develop & Fix
 - Scale out

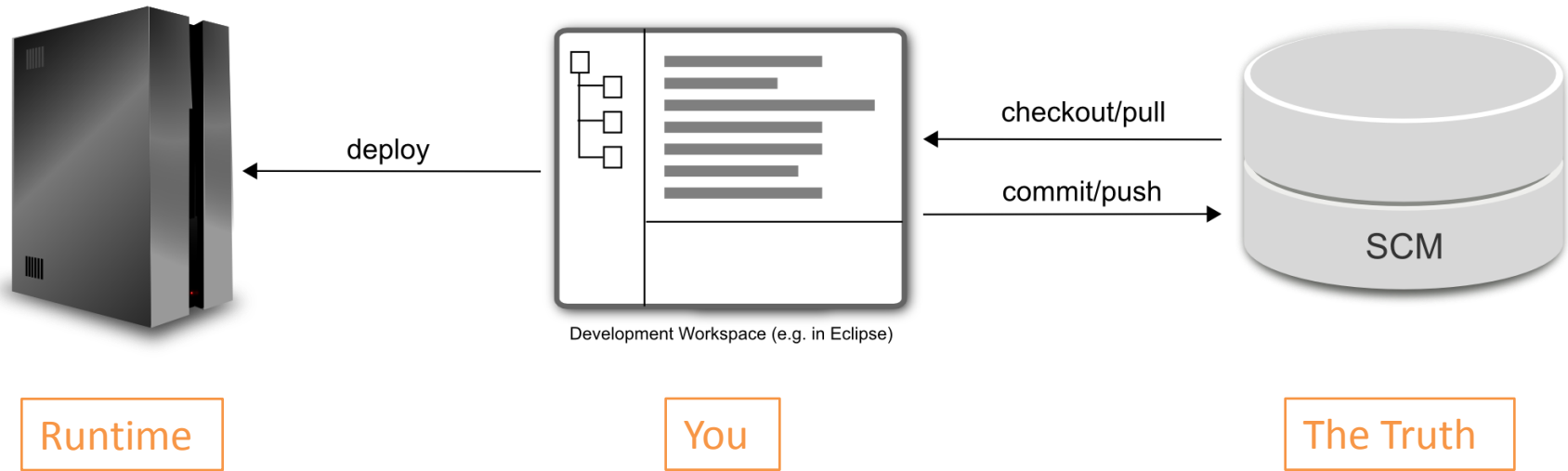
In other words: Improve solution life cycle handling

The Trouble w/ Build Systems

Local Build environments...

- > Are complex to maintain and debug
- > Hold solution artifacts in the wrong place
- > Repeat structural information of the solution
- > Decouple Runtime from Solution definition
- > Make it hard to start something new...
- > Imply non-trivial investment in production infrastructure

Leaking Integration

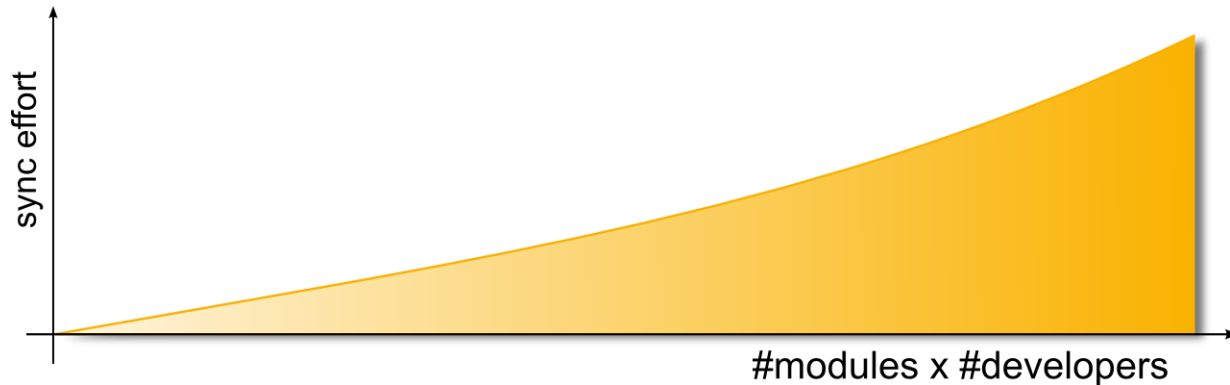


- > Solutions evolve from small and simple to large and modular
- > Solution!=infrastructure => Distributed complexity
- > Sync & Build & Deploy error prone => Developers defer integration
- > Runtime!=Truth => You are on a deadly non-integration vector!

Fixing the Integration Leak

1. Pump!

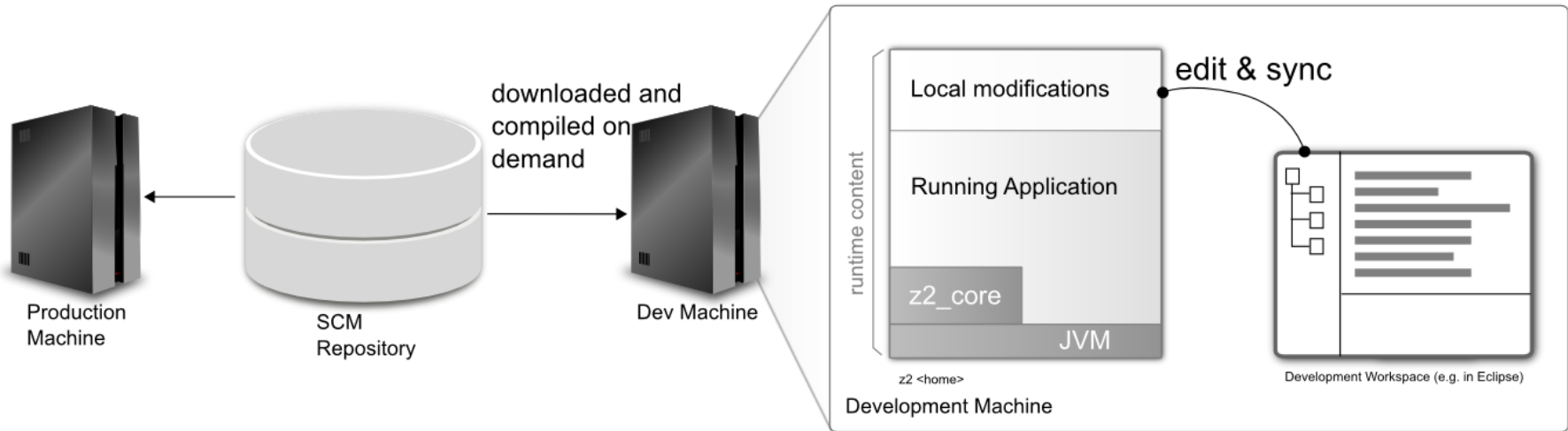
- Add infrastructure on the side to validate and support
- E.g. Continuous Integration w/ ANT or Maven (+Repos) et al
- But the following seems to be the case:



2. Plug the leak!

- Z2-Environment

How does Z2 work



- > The Z2 runtime loads updates directly from the SCM
- > Compilation (e.g. for Java) is done by the runtime when necessary
- > Local override from workspace for local development
- > The whole solution defined in one repository structure == a System

We call it a **System-Centric** approach

Travel Portal: About ~250 kLoJC, 7.5kLoJSPC

DEMO

Z2 Integrations

- > Z2 with Spring
 - Spring for the in-module wiring
 - Z2 for the cross-module wiring
- > Z2 with Hadoop
 - Map/Reduce jobs as integrated components
 - Full cross-module re-use
- > Z2 with Quartz
 - Jobs as integrated components
- > ... You get the point.

What Z2 is not...

- > Z2 is no replacement for Maven
 - Maven is for many smallish independent projects that want to re-use
 - I.e. it's Apache
- > Z2 is not useful for library development
 - Unless it's a tool based on the solution
 - ... Z2 is really for solution development (once more...)
- > Z2 is no build tool
 - It is possible to download binaries
 - Useful for e.g. automated, scheduled Unit Tests

Henning Blohm
ZFabrik Software KG

www.z2-environment.eu
henning.blohm@zfabrik.de

JAZOON'12



SET 2012