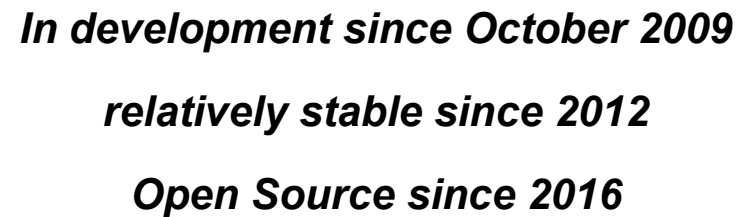




Agenda

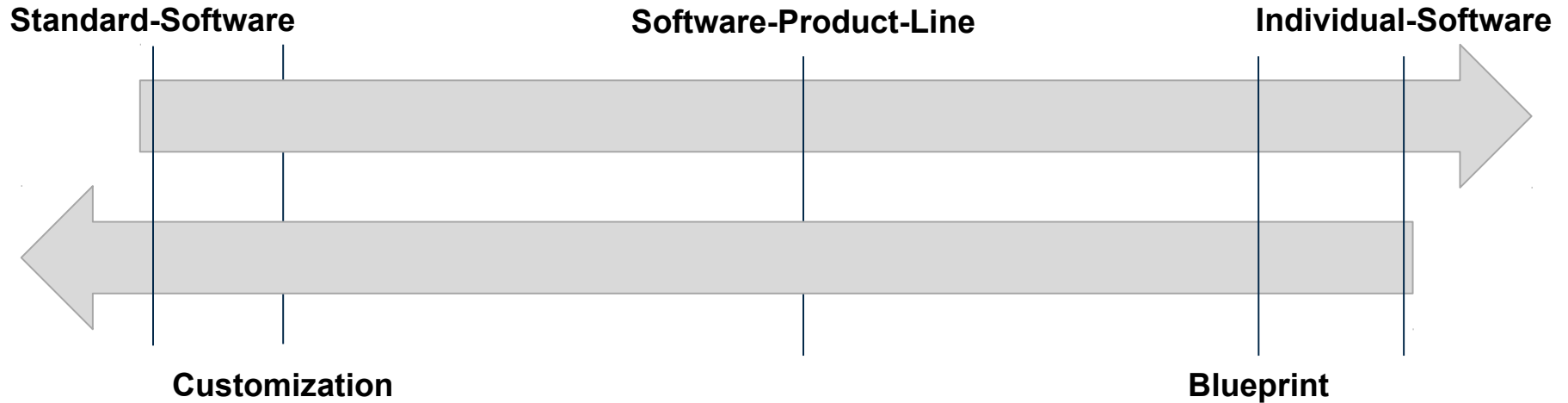
Feel free to ask questions!

Category	Count
FUN	1
Development Comfort	1
Flexibility	1
Reuse	4
Leverage	3





2. Software-Product-Line - Classification -



Reuse

Variability

2. Software-Product-Line

- Assets -

-

- Documentation, Code, Libs, Configuration, Products, Platform

-

-

-

- Synergies

-

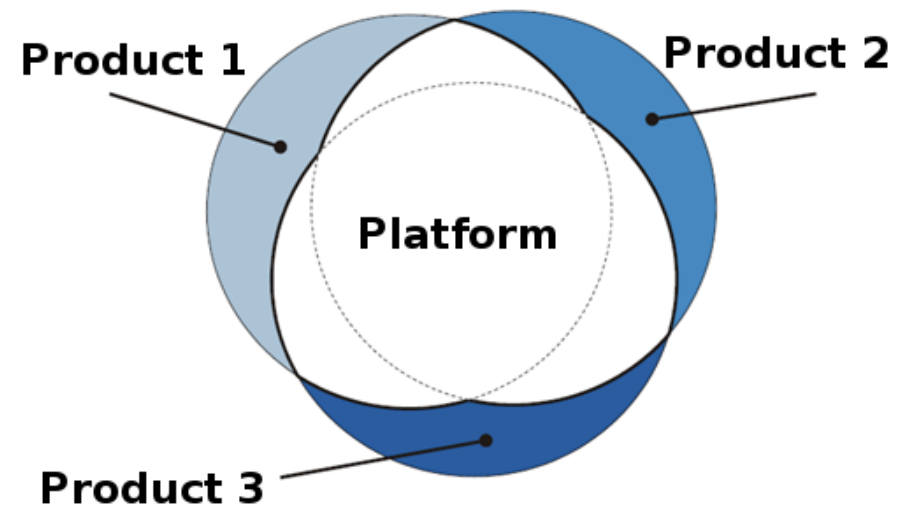
-

-

-

- Bundle Artifacts

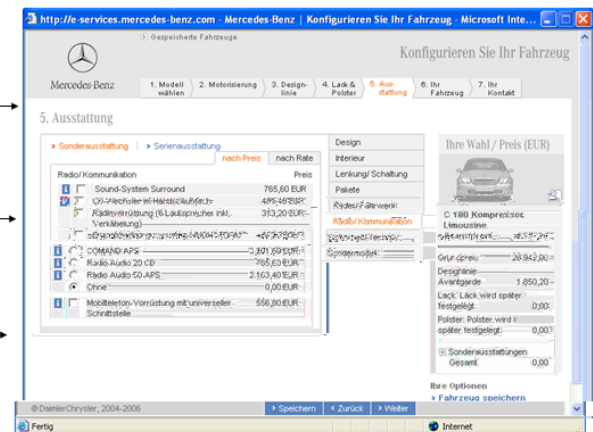
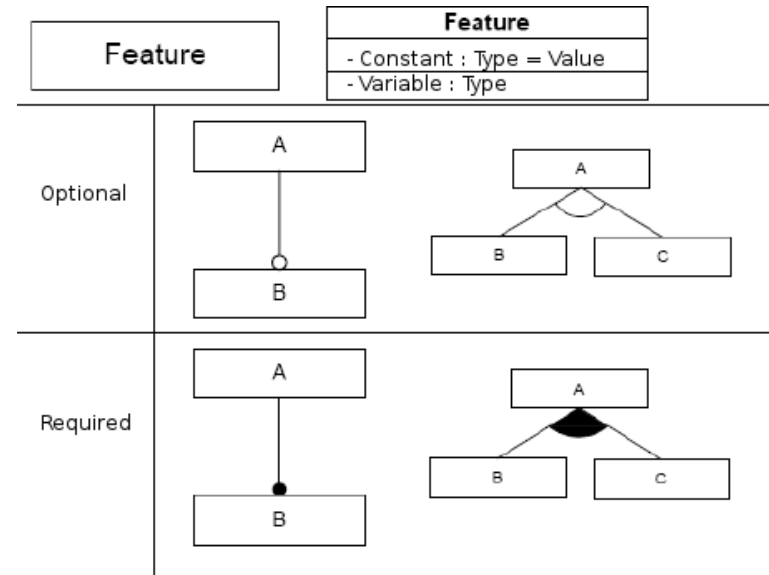
-

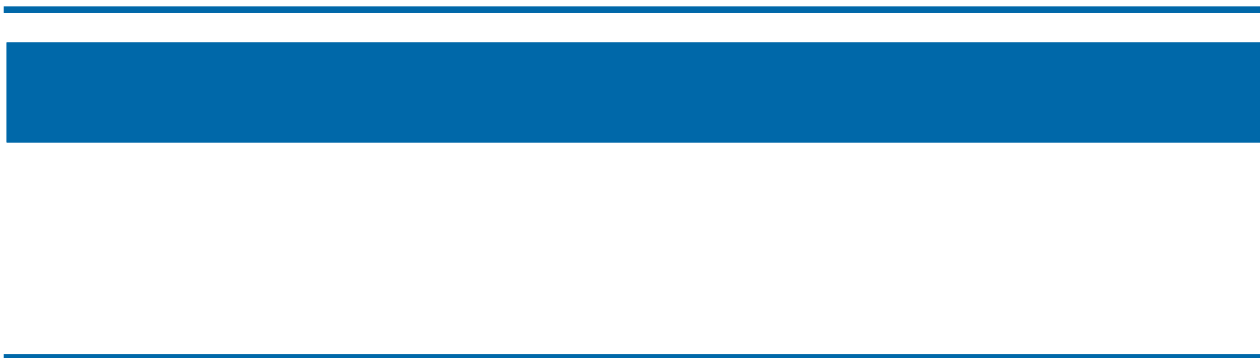


2. Software-Product-Line - Variability -

Features

where





3. Development Environment - Low Impact -

- **OS:**
- **Versioning:**
- **Language:**
- **IDE:**
- **Build & Dependency Mgmt:**

- **Documentation:**



Einrichtung der Entwicklungsumgebung

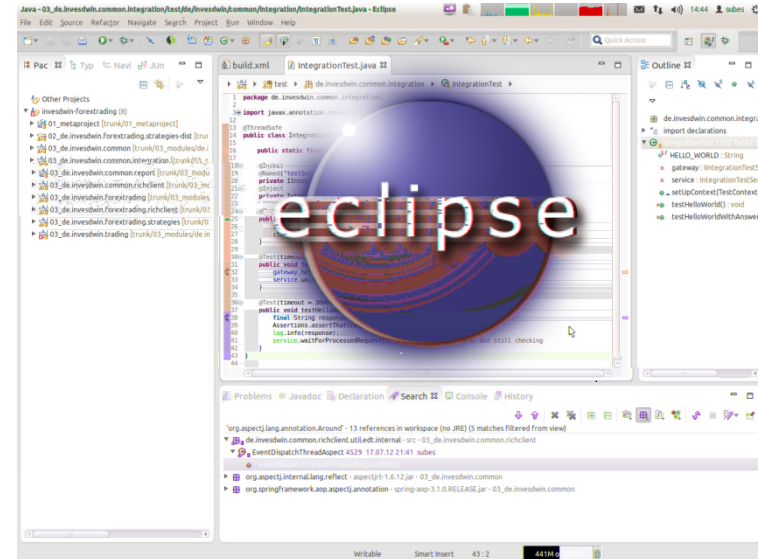
Wir nutzen immer sofern möglich die aktuellste Version der jeweils eingesetzten Software. Die wichtigsten Grundkomponenten sind Java und Eclipse. Wie ein System auf dieser Basis zu konfigurieren ist, wird in diesem Artikel beschrieben. Die Software ist in der Regel so ausgewählt, dass sie sowohl unter Windows als auch Linux als Umgebung zur Verfügung steht. Empfohlen wird jedoch Linux, daher beschreibt dieser Artikel wie man sich Ubuntu als empfohlene Umgebung einrichtet.

Aus folgenden Bestandteilen besteht die Entwicklungsumgebung:

- [illegible]



WARNUNG: Bei dieser Softwareauswahl handelt es sich um Lizenztechnisch unbedenkliche Lösungen. Falls ein Entwickler etwas anderes unter eigener Lizenz verwenden möchte, tut er dies auf eigene Verantwortung und Gefahr.



Plugins

- Eclipse benötigt einige Plugins, um damit arbeiten zu können. In der Regel werden diese Plugins als Dropin angeboten. Empfohlen werden die Plugins auch in genau der Reihenfolge installiert, in der sie aufgelistet sind. Jedes Plugin hat eine Seite, auf der beim jeweiligen Plugin genannt. Falls sich eine Update-Site mit dem Plugin verbinden lässt, kann man sich den aktuellen Link von der Homepage herausfinden. In der Regel sind die Installationsmethoden zur Verfügung steht, sollte dies am besten funktionieren. Falls die Installation über den Eclipse Marketplace nicht funktioniert, ist es möglich, dass das Plugin informiert.

Subversion

moving

Marketplace: Eclipse Marketplace
Modul: Subversive - SVN Team Provide

Konfiguration
Folgende Einstellungen sollte man am Plugin vornehmen:

- Folgende Einstellungen sollte man am Plugin vornehmen:
- 1. Nach der Installation wird man gefragt, ob man das Plugin installieren möchte. **SVN Kit** ist mittlerweile recht ausgereift und sollte sich die aktuellste Version von diesem Konnektor installieren lassen.
 - 2. Unter **Window > Preferences > Team > SVN > Perforce** wählt man **Calculate precise Team Menu enablements** an.
 - 3. Unter **Window > Preferences > Team > SVN > Labels** wählt man **Decorations** an. **Decorations** wählen man alle **Decorations > Text Decorations** und **Decorations > Text Decorations** aus. **author** und **revision** sind ebenfalls zu aktivieren.
 - 4. Unter **Window > Preferences > Team > SVN > Log** wählt man **Decorations** an. **Decorations** wählen man alle **Decorations > Text Decorations** und **Decorations > Text Decorations** aus. **author** und **revision** sind ebenfalls zu aktivieren.
 - 5. Falls ein Projekt nicht automatisch als SVN-Projekt erkannt wird, kann man es mittels **Recht Maustaste auf das Projekt** erzwingen. **SVN** ist dann die richtige Wahl.
 - 6. Falls man das **SVN** Plugin nicht als **SVN** Projekt erkennt, kann man es mittels **Recht Maustaste auf das Projekt** erzwingen. **SVN** ist dann die richtige Wahl.

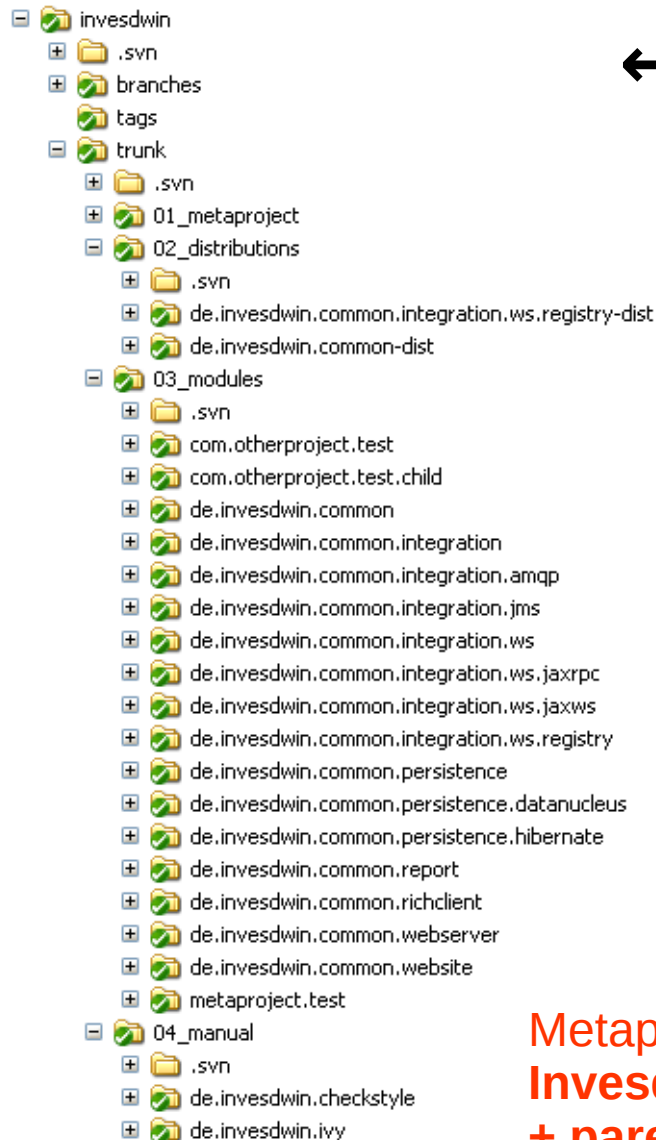
wird, kann man es mittels **Recht Maustaste auf das Projekt** es auch klappen, das Projekt aus dem Workspace zu entfernen.



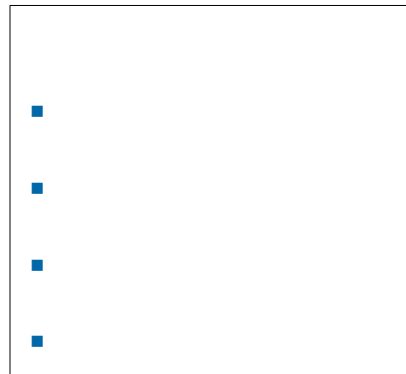
4. Configuration Management - Overview -

REPLACED BY MAVEN

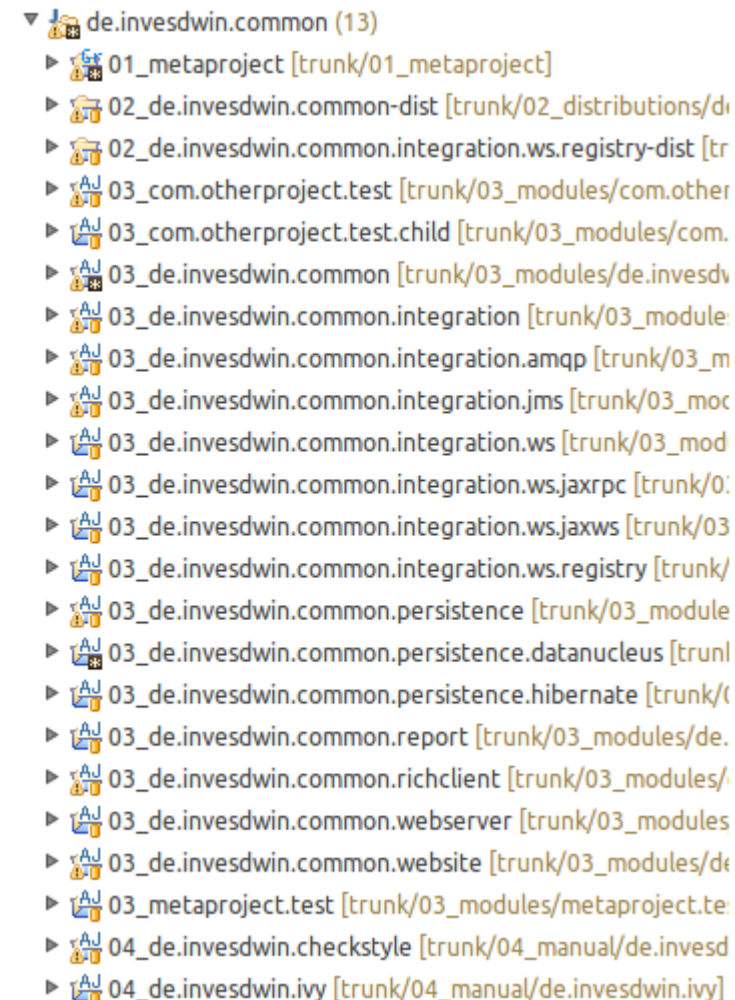
with a better implementation of this concept



← Explorer
Eclipse →



Metaproject got turned into: Invesdwin-maven-plugin + parent pom.xmls



4. Configuration Management

- Module I -

- **Reusable Building Blocks for Products**

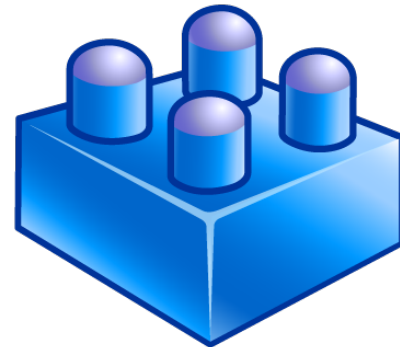
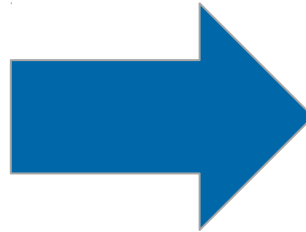
- **Technology:**

- **Domain:**

→ Each has its own



Insurer, Contract, Customer, ...



Contract Management

4. Configuration Management

- Module II -

Previous Platform had a monolithic architecture where services/beans/components were bundled into „containers“ that had tight coupling among each other.

A module in this sense can bundle multiple containers, thus is more than a container.

- - **Module > Container**
 - **Upgrade-Path**
 -
 -



- - **YAGNI & KISS**
 - | | |
|---|-------------------------|
| Green Field | Over-Engineering |
| → Only add Variability when it is actually needed | |
 - **Loose Coupling**
 - Utilize submodules to encapsulate Functionality/Alternatives

4. Configuration Management - Module III -

- Why no OSGi?

-
-
-
-
-

reasoned deviations from the



**only
OSGi
Compatible**

4. Configuration Management

- Distribution -

- A configurable, deployable Product

-
- Configuration of Variability

-
-
-

-
-

-

- Fat-Jar

- Zip

- War

→ Example Web-Application

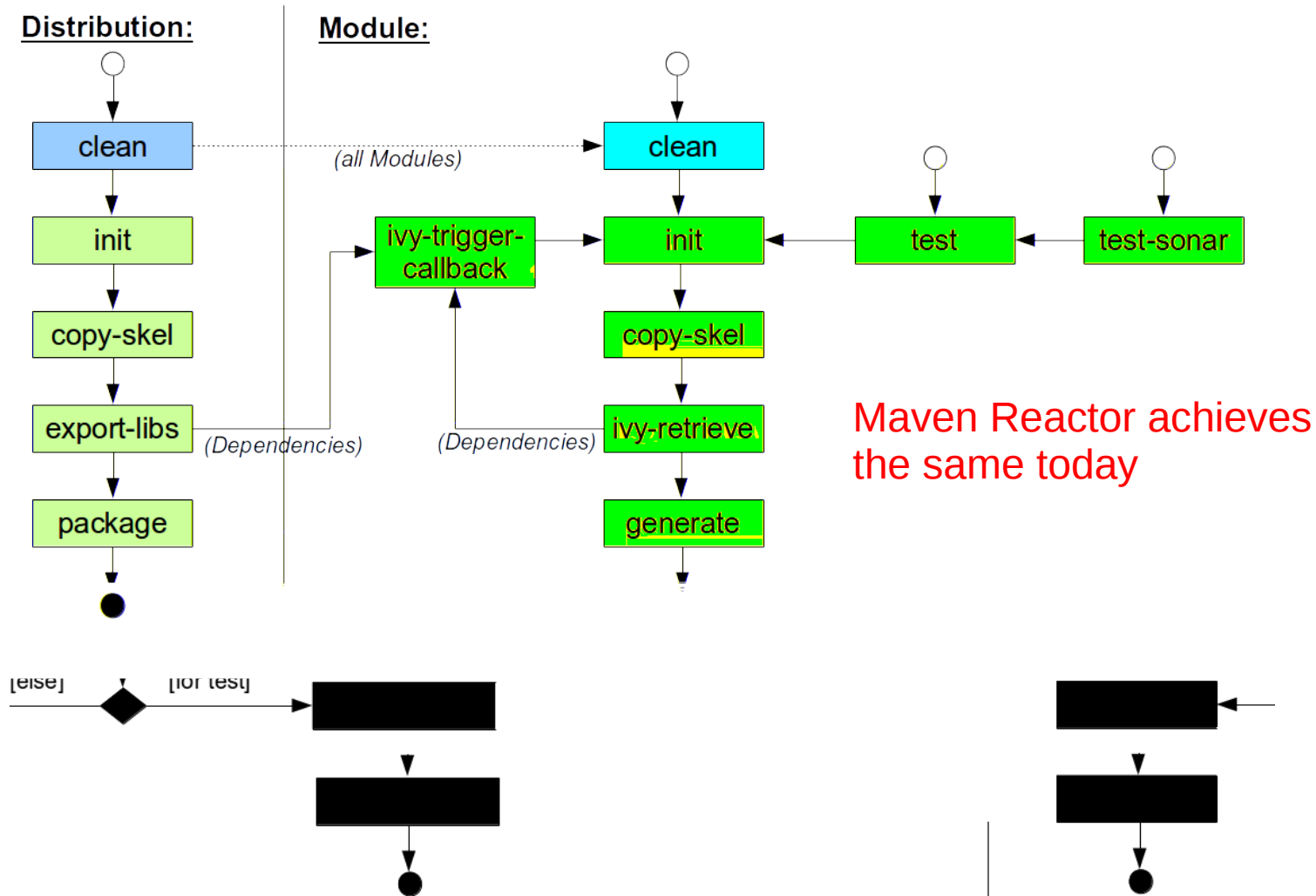


or

4. Configuration Management - Metaproject I -

REPLACED BY MAVEN
with a better implementation of this concept

Definition of unified Build-Process:



4. Configuration Management - Metaproject II -

REPLACED BY MAVEN
with a better implementation of this concept

- Reusable standardized Ant-Scripts
- Project Templates for Modules and Distributions →
- Other Assets inherit Eclipse-Project-Configuration →

The screenshot displays an IDE interface with two main panels. The left panel shows a project tree for '01_metaproject' with subfolders like 'src', 'lib', and 'nbproject'. The right panel shows an 'ivy.xml' file with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:ivy="antlib:org.apache.ivy.ant" name="com.otherproject.test.child" basedir="." default="default">
  <property name="metaproject.dir" location="../01_metaproject"/>
  <import file="${metaproject.dir}/src/build.module.xml"/>
</project>
```

Below the 'ivy.xml' file, a console window shows a Maven build log for '03_de.invesdwin.common.persistence build.xml'. The log includes the following text:

```
<terminated> 03_de.invesdwin.common.persistence build.xml [Ant Build] /usr/lib/jvm/java-1.6.0-openjdk/bin/java (22.08.2012 12:32:54)
basis-copy-skel:
[delete] Deleting directory /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence/.set
[copy] Copying 3 files to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence/.set
[copy] Copying 1 file to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence
[copy] Copying 1 file to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence
[SVN] <Propset> started ...
[SVN] <Propset> finished.
[SVN] <Propset> started ...
[SVN] <Propset> finished.
[SVN] <Propset> started ...
[SVN] <Propset> finished.
copy-skel:
[copy] Copying 1 file to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence
[copy] Copying 1 file to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence
[copy] Copying 1 file to /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence
[delete] Deleting directory /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence/lib
ivy-report:
[ivy:report] Processing /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/01_metaproject/lib/ivy/cache/de.invesdwin-de.invesdwin
[ivy:report] Processing /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/01_metaproject/lib/ivy/cache/de.invesdwin-de.invesdwin
[ivy:report] Processing /home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/01_metaproject/lib/ivy/cache/de.invesdwin-de.invesdwin
BUILD SUCCESSFUL
```

Overlaid on the right panel is the text: **Maven won when it introduced parallel builds**.

4. Configuration Management - Metaproject III -

REPLACED BY MAVEN
with a better implementation of this concept

Automatically generated Eclipse-Classpath for Dependencies:

-
-
-
-

M2E Plugin for Eclipse
achieves the same today

ivy.xml

```
Dashboard build.module.xml ivy.xml build.xml de.invesdwin-de.invesdwin.common.per ivy report :: c
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <ivy-module version="2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="http://ant.apache.org/ivy/schemas/ivy.xsd"
4   xmlns="http://ant.apache.org/ivy/maven">
5   <info>
6     organisation="de.invesdwin"
7     module="${ant.project.name}"
8     status="release"
9     revision="${ivy.version.invesdwin}"
10  </info>
11  <configurations>
12    <conf name="default"/>
13    <conf name="tests" />
14    <conf name="sources" />
15  </configurations>
16  <publications defaultconf="default">
17    <artifact conf="default" type="jar" ext="jar"/>
18    <artifact conf="tests" type="jar" ext="jar" classifier="tests"/>
19    <artifact conf="sources" type="source" ext="jar" classifier="sources"/>
20  </publications>
21  <dependencies>
22    <!-- spring orm -->
23    <dependency org="org.springframework" name="spring-orm" rev="${ivy.version.spring}" conf="default->master;sources" />
24    <dependency org="org.springframework" name="spring-jdbc" rev="${ivy.version.spring}" conf="default->master;sources" />
25    <dependency org="com.mysql.querydsl" name="querydsl-jpa" rev="${ivy.version.querydsl}" conf="default->master;sources" />
26
27    <!-- connection pooling -->
28    <dependency org="com.jolbox" name="bonecp" rev="${ivy.version.bonecp}" conf="default->master;sources" />
29
30    <!-- jpa metamodel generator -->
31    <dependency org="org.hibernate" name="hibernate-jpamodelgen" rev="${ivy.version.hibernate-jpamodelgen}" conf="default->master;sources" />
32
33    <!-- Test Konnektoren für Datenbanken -->
34    <dependency org="com.h2database" name="h2" rev="${ivy.version.h2}" conf="tests->master;sources" />
35    <dependency org="mysql" name="mysql-connector-java" rev="${ivy.version.mysql-connector}" conf="tests->master;sources" />
36    <dependency org="de.invesdwin" name="de.invesdwin.common" rev="${ivy.version.invesdwin}" conf="default;tests;sources" />
37  </dependencies>
38 </ivy-module>
```

Dashboard build.module.xml ivy.xml build.xml de.invesdwin-de.invesdwin.common.per ivy report :: de.invesdwin.common.pe

file:///home/subes/Dokumente/Entwicklung/invesdwin/inv-common/trunk/03_modules/de.invesdwin.common.persistence/lib/1/de.invesdwin-de.invesdwin.common.persistence

de.invesdwin.common.persistence 0.2.0 by de.invesdwin

resolved on 2012-08-22 12:32:58

default tests sources

Dependencies Stats

Modules: 59 (0 downloaded, 0 failed)
59 (0 downloaded, 0 failed)
14252 kB (0 kB downloaded, 14252 kB in cache)

Dependencies Overview

Module	Revision	Status	Resolver	Default	Licenses	Size
common by	0.2.0	release	temp	false		244 kB
modelgen by	1.1.1.Final	release	invesdwin	false	Apache License, Version 2.0	157 kB
n.jolbox	0.7.2-SNAPSHOT	integration	repo	false		116 kB
y com.mysena.querydsl	2.3.0	release	invesdwin	false		94 kB
org.springframework	3.2.0.M1	release	invesdwin	false	The Apache Software License, Version 2.0	374 kB
r.org.springframework	3.2.0.M1	release	invesdwin	false	The Apache Software License, Version 2.0	394 kB

Details

common by de.invesdwin

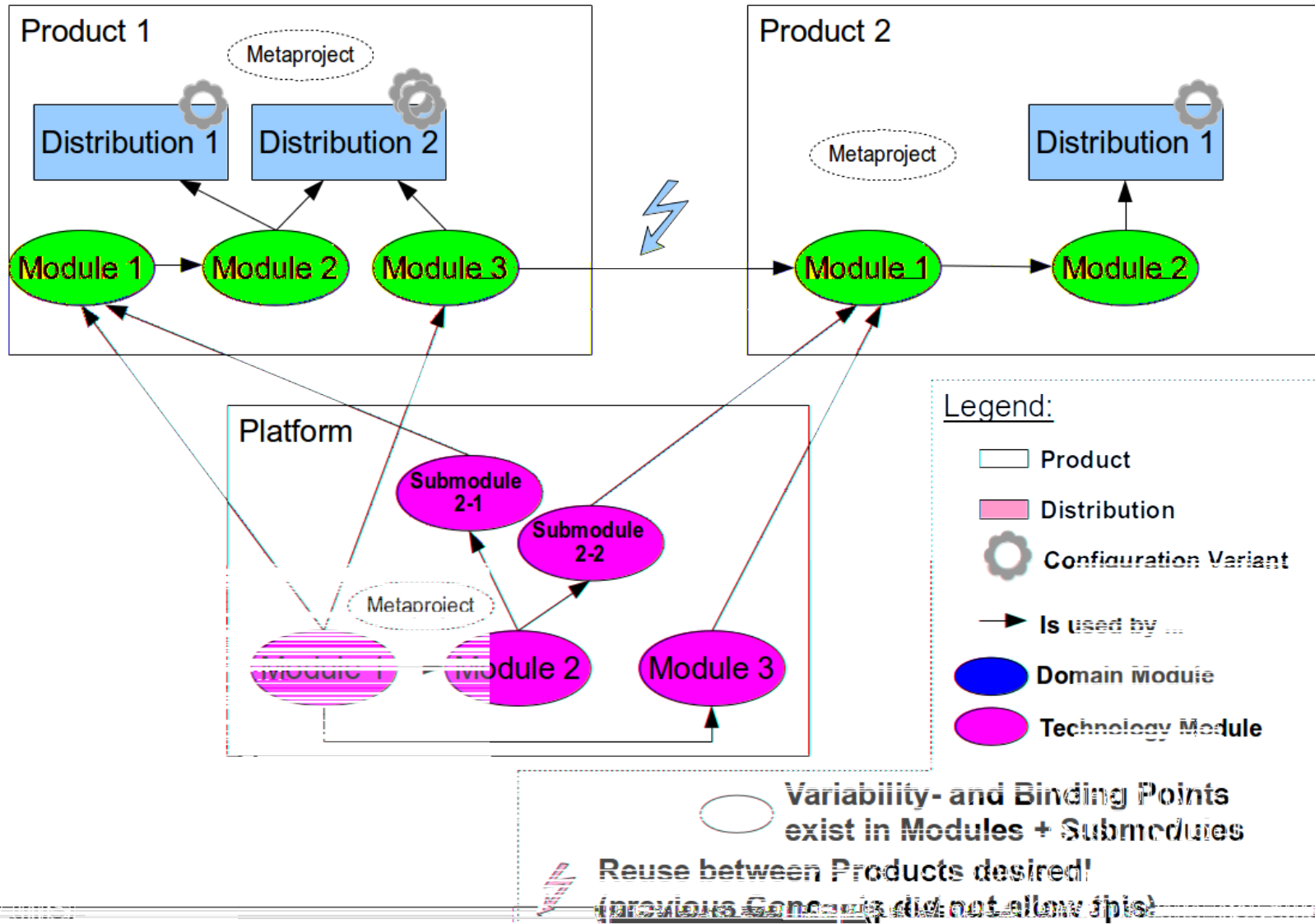
Revision	Status	Publication
20120809132657	release	

de.invesdwin.

Revision: 0.2.0

Status	Publication
release	

4. Configuration Management - Multi-Product-Management -



5. Assets - Platform I -

Outdated Information

some components/names have changed
also today there are more modules

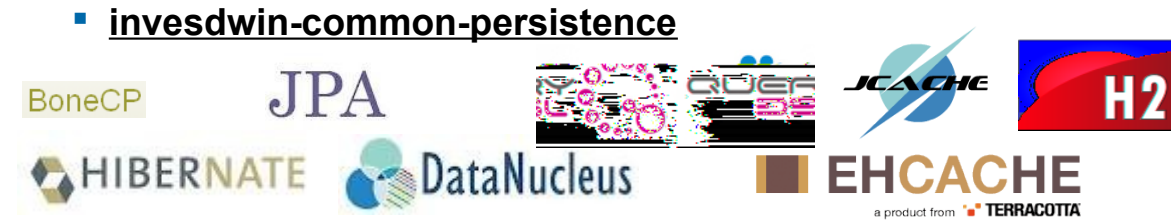
metaproject



investdwin-common



investdwin-common-persistence



investdwin-common-report



investdwin-common-richclient



Infrastructure



Dependency Repos



Google WindowBuilder Pro



■

5. Assets

- Example Product webproxy -



HtmlUnit

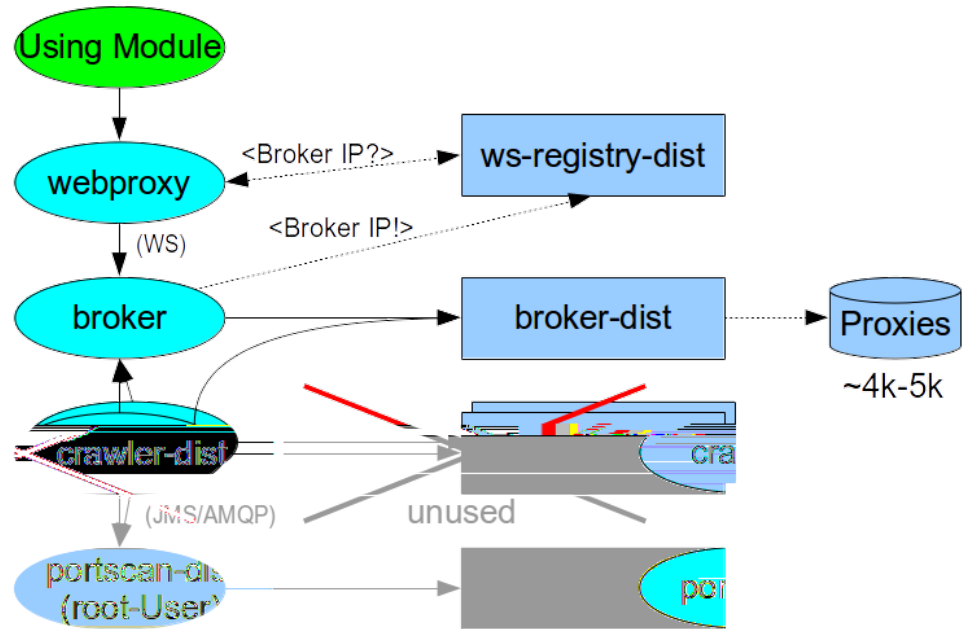
GUI-Less browser for Java programs



JAKARTA COMMONS
HTTP CLIENT



win/lib/j - *Peap*



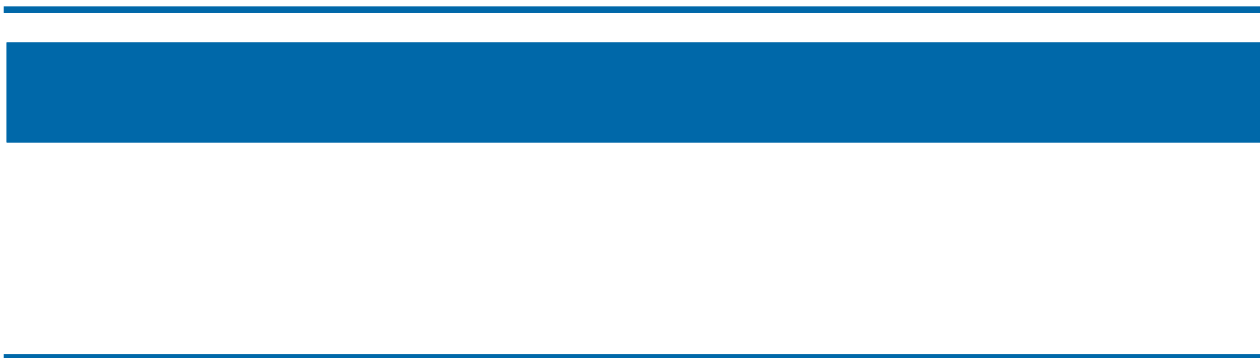
Legend:

Is used by ...

Technology Module

Distribution

Database



- Application Bootstrap -

```
Problems Javadoc Declaration Search Console Progress History SVN Properties
<terminated> 03_de.invesdwin.common [JUnit] /usr/lib/jvm/java-1.6.0-openjdk/bin/java (08.08.2012 18:55:17)
2012-08-08 18:55:21,369 [main] INFO d.i.common.beans.context.PreMergedContext.logLogbackBeingCon - Loading 2 logback configs from classpath [/META-INF/logback/common.logback.x
2012-08-08 18:55:21,740 [main] INFO de.invesdwin.common.CommonProperties.info - Loading 1 base package in classpath [de.invesdwin]
2012-08-08 18:55:22,479 [main] INFO d.i.common.beans.context.MergedContext.logContextsBeingLoade - Loading 1 spring context from classpath in TEST config [/META-INF/ctx.spring
2012-08-08 18:55:23,122 [main] INFO d.i.c.b.c.internal.EhCacheConfigurationMerger.logXmIsBeingLo - Loading 1 ehcache config from classpath [/META-INF/ehcache/default.ehcache.x
2012-08-08 18:55:23,634 [main] INFO d.i.common.beans.context.MergedContext.info - Bootstrap finished after: PT3.411.182.527S
2012-08-08 18:55:23,701 [main] INFO de.invesdwin.common.beans.context.TestTest.tearDown - 1. test finished after: PT0.023.404.931S
2012-08-08 18:55:23,730 [main] INFO de.invesdwin.common.beans.context.TestTest.tearDown - 2. test finished after: PT0.000.061.740S
2012-08-08 18:55:23,740 [main] INFO de.invesdwin.common.beans.context.TestTest.tearDown - 3. test finished after: PT0.000.041.905S
2012-08-08 18:55:23,900 [main] INFO d.i.common.beans.context.MergedContext.logContextsBeingLoade - Loading 1 spring context from classpath in TEST config [/META-INF/ctx.spring
2012-08-08 18:55:24,076 [main] INFO d.i.c.b.c.internal.EhCacheConfigurationMerger.logXmIsBeingLo - Loading 1 ehcache config from classpath [/META-INF/ehcache/default.ehcache.x
2012-08-08 18:55:24,119 [main] INFO de.invesdwin.common.lang.time.DurationTest.tearDown - 4. test finished after: PT0.000.099.733S
2012-08-08 18:55:24,134 [main] INFO de.invesdwin.common.lang.time.DurationTest.tearDown - 5. test finished after: PT0.010.332.317S
2012-08-08 18:55:24,139 [main] INFO de.invesdwin.common.lang.time.DurationTest.tearDown - 6. test finished after: PT0.000.065.930S
2012-08-08 18:55:24,172 [main] INFO de.invesdwin.common.lang.time.DurationTest.tearDown - 7. test finished after: PT0.027.914.443S
2012-08-08 18:55:24,269 [main] INFO d.i.common.beans.context.MergedContext.logContextsBeingLoade - Loading 1 spring context from classpath in TEST config [/META-INF/ctx.spring
2012-08-08 18:55:24,411 [main] INFO d.i.c.b.c.internal.EhCacheConfigurationMerger.logXmIsBeingLo - Loading 1 ehcache config from classpath [/META-INF/ehcache/default.ehcache.x
2012-08-08 18:55:24,601 [main] INFO de.invesdwin.common.beans.context.MockitoTest.tearDown - 8. test finished after: PT0.000.070.121S
2012-08-08 18:55:24,700 [main] INFO d.i.commonr.beans.context.MergedContext.logContextsBeingLoade - Loading 1 spring context from classpath in TEST config [/META-INF/ctx.spring
2012-08-08 18:55:24,854 [main] INFO d.i.c.b.c.internal.EhCacheConfigurationMerger.logXmIsBeingLo - Loading 1 ehcache config from classpath [/META-INF/ehcache/default.ehcache.x
2012-08-08 18:55:24,950 [main] INFO d.i.common.Lang.math.scaled.PercentTest.tearDown - 9. test finished after: PT0.054.173.645S
```

6. Binding Variability

- Properties I -

■ All Properties are System-Properties

■ available

-
-
-
-

```
PortscanProperties.java
package de.invesdwin.webproxy.portscan.internal;

import java.net.InetAddress;

public class PortscanProperties {

    public static final InetAddress CHECK_HOST;
    public static final int CHECK_PORT = 80;
    public static final int LOCAL_BIND_PORT;
    public static final Duration ICMP_RESPONSE_TIMEOUT;

    public static final SystemProperties SYSTEM_PROPERTIES = new SystemProperties(PortscanProperties.class);

    public {
        CHECK_HOST = NetworkUtil.toAddress(SYSTEM_PROPERTIES.getString("CHECK_HOST"));
        LOCAL_BIND_PORT = readLocalBindPort();
        ICMP_RESPONSE_TIMEOUT = SYSTEM_PROPERTIES.getDuration("ICMP_RESPONSE_TIMEOUT");
    }

    public PortscanProperties() {}

    public final String key = "LOCAL_BIND_PORT";
    public final Integer value = SYSTEM_PROPERTIES.getInt(key);
    Assertions.assertThat(NetworkUtil.isPort(value))
        .isTrue();
    return value;
}
```

```
de.invesdwin.webproxy.portscan.properties
1 #On the host port 80 must be open and a service has to be running on it. The host also has to answer pings so that the check
2 de.invesdwin.webproxy.portscan.internal.PortscanProperties.CHECK_HOST=google.de
3 de.invesdwin.webproxy.portscan.internal.PortscanProperties.LOCAL_BIND_PORT=44125
4 de.invesdwin.webproxy.portscan.internal.PortscanProperties.ICMP_RESPONSE_TIMEOUT=3 SECONDS
5 #For timings see: http://www.networkuptime.com/nmap/page09-09.shtml
6 de.invesdwin.webproxy.portscan.internal.PortscanProperties.UPLOAD_PAUSE_BETWEEN_PACKETS=0 MILLISECONDS
7 de.invesdwin.webproxy.portscan.internal.PortscanProperties.UPLOAD_PAUSE_BETWEEN_PACKETS_PER_HOST=0 MILLISECONDS
8 de.invesdwin.webproxy.portscan.internal.PortscanProperties.RESPONSE_TIMEOUT_BETWEEN_SYN_PACKETS_PER_HOST=500 MILLISECONDS
9 de.invesdwin.webproxy.portscan.internal.PortscanProperties.MAX_OPEN_ICMP_REQUESTS=25
10 de.invesdwin.webproxy.portscan.internal.PortscanProperties.MAX_OPEN_SYN_REQUESTS=10
```

```
ctx.persistence.test.memory.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3       xmlns:context="http://www.springframework.org/schema/context"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans
5                           http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="persistenceProperties" class="de.invesdwin.common.system.properties.SystemPropertiesDefinition">
8         <property name="systemProperties">
9             <map>
10                 <entry key="javax.persistence.jdbc.driver" value="org.h2.Driver" />
11                 <entry key="javax.persistence.jdbc.url" value="jdbc:h2:mem:invesdwin;DB_CLOSE_ON_EXIT=FALSE" />
12                 <entry key="javax.persistence.jdbc.user" value="sa" />
13                 <entry key="javax.persistence.jdbc.password" value="sa" />
14                 <entry key="hibernate.dialect" value="org.hibernate.dialect.H2Dialect" />
15                 <entry key="hibernate.hbm2ddl.auto" value="create" />
16             </map>
17         </property>
18     </bean>
19
20     <import resource="actx.persistence.hibernate.xml" />
21
22 </beans>
```

Applications

Start Page x de.invesdwin.webproxy.broker-prod-0.2.0.jar (pid 15284) x

de.invesdwin.webproxy.broker-prod-0.2.0.jar (pid 15284)

Overview

PID: 15284
Host: localhost
Main class: de.invesdwin.webproxy.broker-prod-0.2.0.jar
Arguments: <none>

JVM: OpenJDK 64-Bit Server VM (22.0-b10, mixed mode)
Java: version 1.7.0_03, vendor Oracle Corporation
Java Home: /usr/lib/jvm/java-7-openjdk-amd64/jre
JVM Flags: <none>

Heap dump on OOME: disabled

JVM arguments System properties

de.invesdwin.webproxy.WebproxyProperties.PROXY_POOL_COOLDOWN_MIN_TIMEOUT=100 MILLISECONDS
de.invesdwin.webproxy.WebproxyProperties.PROXY_POOL_WARMUP_TIMEOUT=10 MINUTES
de.invesdwin.webproxy.WebproxyProperties.PROXY_VERIFICATION_REDIRECT_SLEEP=15 SECONDS
de.invesdwin.webproxy.WebproxyProperties.PROXY_VERIFICATION_RETRY_ON_ALL_EXCEPTIONS=false
de.invesdwin.webproxy.broker.internal.BrokerProperties.ADDITIONAL_RANDOM_TO_BE_SCANNED_PORTS_PERCENT=25
de.invesdwin.webproxy.broker.internal.BrokerProperties.MAX_SPECIFIC_TO_BE_SCANNED_PORTS=1000
de.invesdwin.webproxy.broker.internal.BrokerProperties.PROXY_DOWNTIME_TOLERANCE=18 HOURS
de.invesdwin.webproxy.crawler.internal.CrawlerProperties.RANDOM_SCAN_ALLOWED=false
de.invesdwin.webproxy.crawler.internal.CrawlerProperties.WAIT_FOR_PORTSCAN_PROCESSING_END=true
de.invesdwin.webproxy.geolocation.internal.GeolocationProperties.GEOIP_DATA_URL=http://geolite.maxmind.com/download/geoipt
de.invesdwin.webproxy.geolocation.internal.GeolocationProperties.GEONAMES_DATA_URL=http://download.geonames.org/export

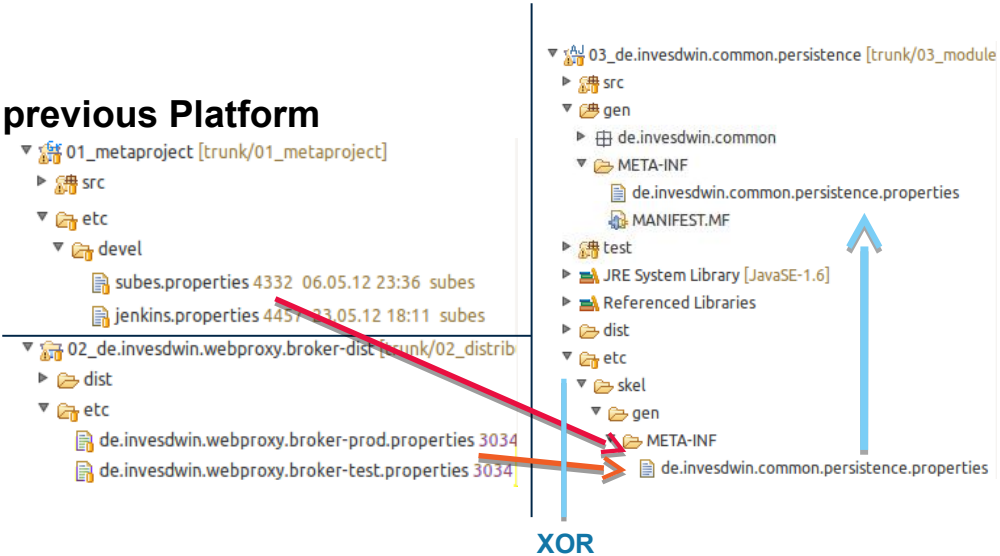
6. Binding Variability - Properties II -

Now per Module Overrides
with Maven Implementation

Developer- and Distribution-Properties as in previous Platform

replaces the

gen-Source
Jar



Variable Binding-Time

	flexibility	performance	code size	complexity
source time	-	+	+	-
compile time	+	+	+	-
link time	+	+	+	-
load time	+++	+	+	+
run time	+++	-	-	+

6. Binding Variability

- Contract Modules -

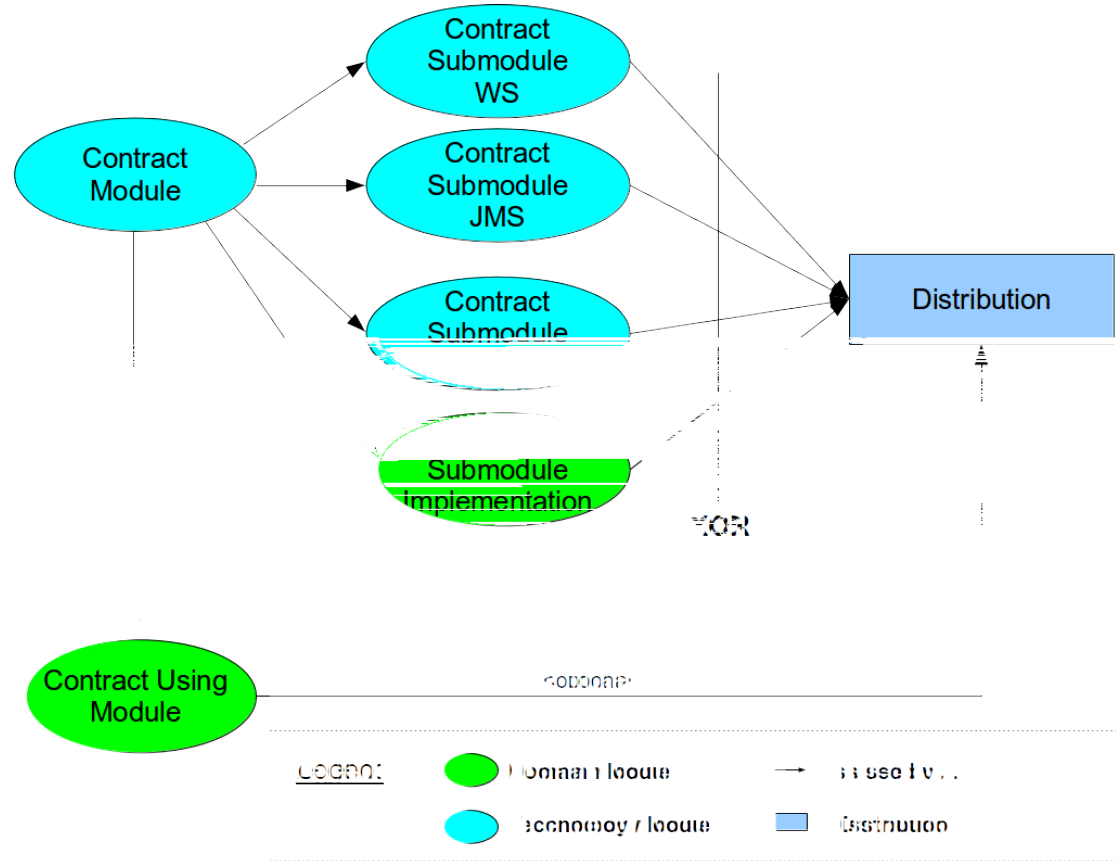
- Book:
- Framework:
- SPL-Usage:

-
-
-

- Contract:
- Contract Submodule:
- Contract Impl:

- Possibilities:

-
-





7. DDD und MDA

- Concepts -

- **DDD – Domain Driven Design/Development**

-
-
-

- **MDA – Model Driven Architecture**

- -
 -
 -
 -
- -
 -
 -
 -
 -





8. Conclusion

- Your opinion is very welcome! -

[illegible]

