

# Software variability management

Xavier Devroey <[x.d.m.devroey@tudelft.nl](mailto:x.d.m.devroey@tudelft.nl)>

Office 4.W.740 (4th floor, West side, VMB building 28)



# Introduction

# Variability in hardware



	Samsung <b>Galaxy S6 Edge</b>	Samsung <b>Galaxy S6</b>	HTC <b>M9</b>	Google/Motorola <b>Nexus 6</b>	Apple <b>iPhone 6 Plus</b>	Apple <b>iPhone 6</b>	Motorola <b>Moto X</b>	Sony <b>Xperia Z3</b>	LG <b>G3</b>
Display size (diagonal)	5.1 in	5.1 in	5.0 in	6.0 in	5.5 in	4.7 in	5.2 in	5.2 in	5.5 in
Resolution (pixels)	577ppi	577ppi	441ppi	493ppi	401ppi	326ppi	423ppi	424ppi	538ppi
Weight	132g	138g	157g	184g	172g	129g	144g	152g	149g
Size, in mm (H x W x D)	142.1 x 70.1 x 7.0	143.4 x 70.5 x 6.8	144.6 x 69.7 x 9.6	159.3 x 83 x 10.1	158 x 77.8 x 7.1	138 x 67 x 6.9	140.8 x 72.4 x 9.9	146 x 72 x 7.3 (mm)	146.3 x 74.6 x 8.9
Operating System	Android 5	Android 5	Android 5	Android 5	iOS 8	iOS 8	Android 5	Android 4.4.4	Android 4.4.2
RAM	3GB	3GB	3GB	3GB	1GB	1GB	2GB	3GB	2/3GB
Processor	2.1/1.5GHz, octa-core	2.1/1.5GHz, octa-core	2.1/1.5GHz, octa-core	2.7GHz quad-core	1.4GHz, dual-core	1.4GHz, dual-core	2.5GHz quad-core	2.5GHz quad-core	2.5Ghz quad-core
Internal storage	32/64/128GB	32/64/128GB	32GB	32/64GB	16/64/128GB	16/64/128GB	16/32GB	16/32GB	16/32GB
Storage expansion	none	none	Up to 128GB, MicroSD	none	none	none	none	Up to 128GB, MicroSD	Up to 128GB, MicroSD
Camera (rear/front)	16MP/5.0MP	16MP/5.0MP	20.7MP/4.0MP	13MP/2MP	8MP/1.2MP	8MP/1.2MP	13MP/2MP	20.7MP/2.2MP	13MP/2.1 MP
Video recording	2160p at 30fps	2160p at 30fps	1080p at 30fps	2160 at 30fps	1080p at 60 fps	1080p at 60 fps	2160p at 30fps	2160p at 30fps	2160p at 30fps
Battery life/ capacity	2600 mAh	2550 mAh	2840 mAh	3220 mAh	2915 mAh	1810 mAh	2300 mAh	3100 mAh	3000 mAh
NFC	yes	yes	yes	yes	yes	yes	yes	yes	yes
Wireless charging	yes	yes	no	yes	no	no	no	no	yes
Water/dust resistant	no	no	no	yes	no	no	yes	yes	no
Launch date	April 2015	April 2015	March 2015	October 2014	September 2014	September 2014	October 2014	September 2014	May 2014

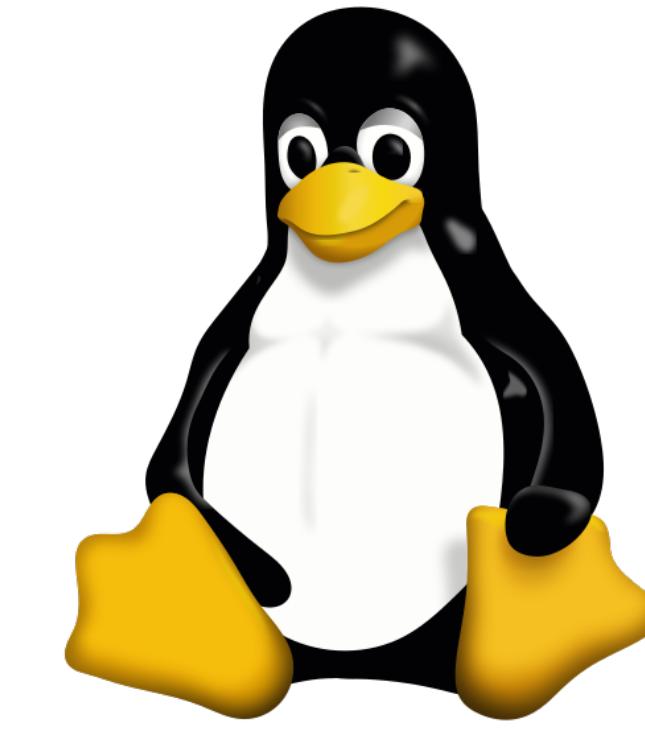
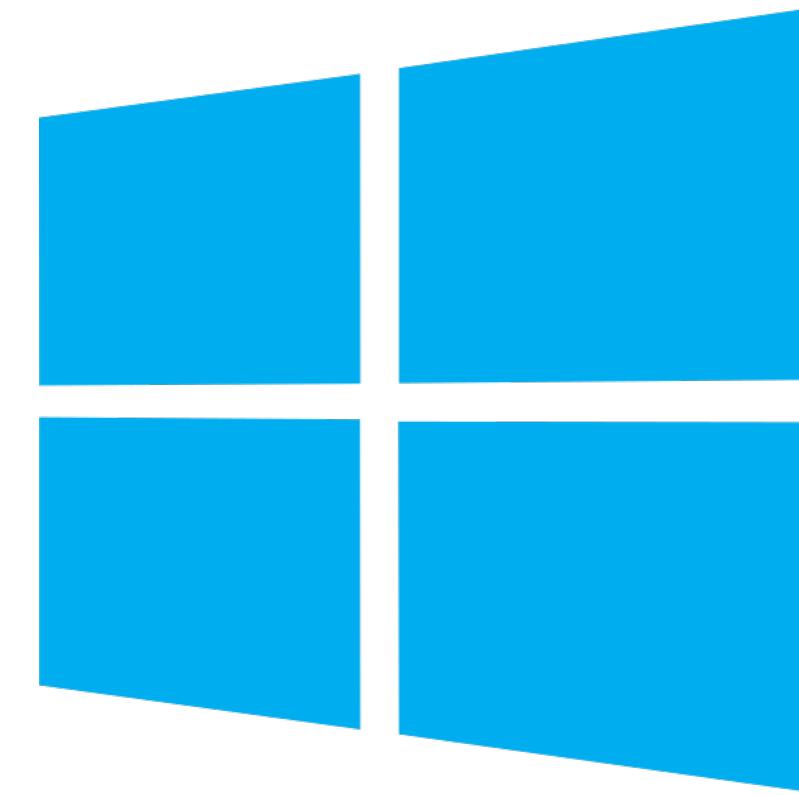
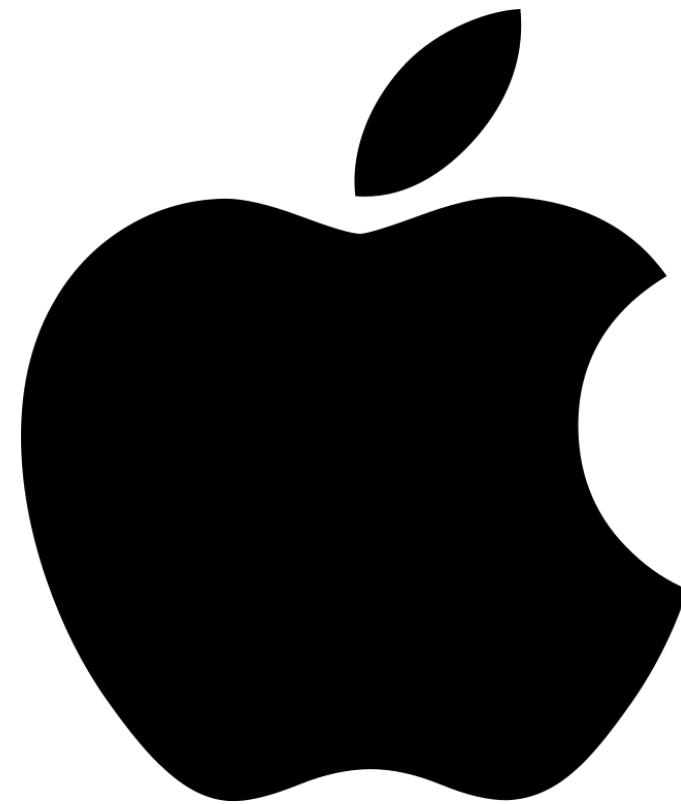
Source: Companies. \*Not officially.

C. Inton, 02/03/2015

REUTERS

# Variability in platforms

Operating System:



Database Management System:



# Variability in software



## - Bundles

	Java EE	Java	C/C++	PHP	Committees	JavaScript/Web	Java/DSL	Modeling Tools	RCP/RAP	Parallel Application	Testers	Scout
Select packages to compare	<input type="checkbox"/>											
BIRT Framework												
C/C++ Development Tools			✓									✓
Data Tools Platform	✓											
Git integration for Eclipse	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
Java Development Tools	✓	✓				✓			✓	✓		✓
Java EE Developer Tools		✓										
JavaScript Development Tools		✓				✓						
Jubula Functional Testing												✓
Maven Integration for Eclipse	✓	✓				✓			✓			✓
Mylyn Task List	✓	✓	✓	✓	✓	✓			✓		✓	✓
PHP Development Tools (PDT)						✓						
Parallel Tools Platform												✓
Plug-in Development Environment	✓					✓			✓	✓		
Remote System Explorer	✓		✓				✓					

<https://www.eclipse.org>

# Variability in software

- Bundles
- *Command line parameters*

gcc -help

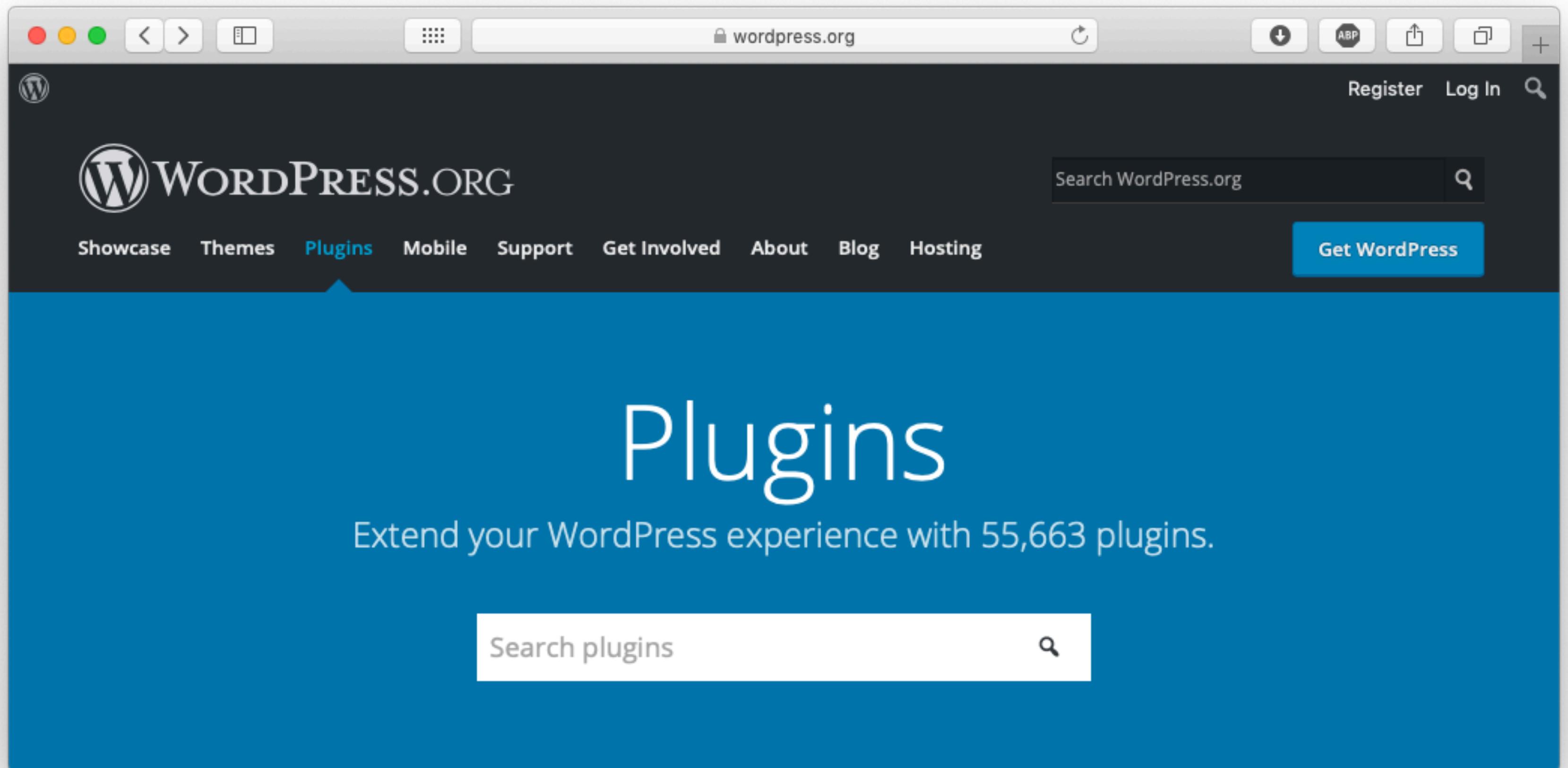
```
OVERVIEW: clang LLVM compiler

USAGE: clang [options] <inputs>

OPTIONS:
-###          Print (but do not run) the commands to run for this compilation
--analyzer-output <value>
               Static analyzer report output format (html|plist|plist-multi|file|plist-html|text).
--analyze      Run the static analyzer
--arcmt-migrate-emits-errors
               Emit ARC errors even if the migrator can fix them
--arcmt-migrate-report-output <value>
               Output path for the plist report
-B <dir>     Add <dir> to search path for binaries and object files used implicitly
-CC           Include comments from within macros in preprocessed output
-cfguard     Emit tables required for Windows Control Flow Guard.
-cl-denorms-are-zero OpenCL only. Allow denormals to be flushed to zero.
-cl-fast-relaxed-math OpenCL only. Sets -cl-finite-math-only and -cl-unsafe-math-optimizations, and defines __FAST_RELAXED_MATH__.
-cl-finite-math-only OpenCL only. Allow floating-point optimizations that assume arguments and results are not NaNs or +Inf.
-cl-fp32-correctly-rounded-divide-sqrt
               OpenCL only. Specify that single precision floating-point divide and sqrt used in the program source are correctly rounded.
-cl-kernel-arg-info OpenCL only. Generate kernel argument metadata.
-cl-mad-enable   OpenCL only. Allow use of less precise MAD computations in the generated binary.
-cl-no-signed-zeros OpenCL only. Allow use of less precise no signed zeros computations in the generated binary.
-cl-opt-disable  OpenCL only. This option disables all optimizations. By default optimizations are enabled.
-cl-single-precision-constant
               OpenCL only. Treat double precision floating-point constant as single precision constant.
-cl-std=<value> OpenCL language standard to compile for.
-cl-strict-aliasing OpenCL only. This option is added for compatibility with OpenCL 1.0.
-cl-uniform-work-group-size
               OpenCL only. Defines that the global work-size be a multiple of the work-group size specified to clEnqueueNDRangeKernel
-cl-unsafe-math-optimizations
               OpenCL only. Allow unsafe floating-point optimizations. Also implies -cl-no-signed-zeros and -cl-mad-enable.
--config <value> Specifies configuration file
--cuda-compile-host-device
               Compile CUDA code for both host and device (default). Has no effect on non-CUDA compilations.
--cuda-device-only Compile CUDA code for device only
--cuda-gpu-arch=<value> CUDA GPU architecture (e.g. sm_35). May be specified more than once.
--cuda-host-only  Compile CUDA code for host only. Has no effect on non-CUDA compilations.
--cuda-include-ptx=<value>
               Include PTX for the following GPU architecture (e.g. sm_35) or 'all'. May be specified more than once.
--cuda-noot-device-debug
               Enable device-side debug info generation. Disables ptxas optimizations.
--cuda-path-ignore-env Ignore environment variables to detect CUDA installation
--cuda-path=<value> CUDA installation path
-cxx-isystem <directory>
               Add directory to the C++ SYSTEM include search path
-C             Include comments in preprocessed output
-c             Only run preprocess, compile, and assemble steps
-dD            Print macro definitions in -E mode in addition to normal output
-dependency-dot <value> Filename to write DOT-formatted header dependencies to
-dependency-file <value>
               Filename (or -) to write dependency output to
-dI            Print include directives in -E mode in addition to normal output
-dM            Print macro definitions in -E mode instead of normal output
-D <macro>=<value> Define <macro> to <value> (or 1 if <value> omitted)
-emit-ast     Emit Clang AST files for source inputs
-emit-llvm    Use the LLVM representation for assembler and object files
-enable-trivial-auto-var-init-zero-knowing-it-will-be-removed-from-clang<value>
               Trivial automatic variable initialization to zero is only here for benchmarks, it'll eventually be removed, and I'm OK with that because I'm only using it to
benchmark
-E             Only run the preprocessor
-faddrsig     Emit an address-significance table
-faligned-allocation Enable C++17 aligned allocation functions
-fallow-editor-placeholders
               Treat editor placeholders as valid source code
-fansi-escape-codes Use ANSI escape codes for diagnostics
-fapinotes-cache-path=<directory>
               Does nothing; API notes are no longer cached separately from modules
-fapinotes-modules Enable module-based external API notes support
```

# Variability in software

- Bundles
- Command line parameters
- *Plugins*



# Variability in software

- Bundles
- Command line parameters
- Plugins
- *Configuration files*

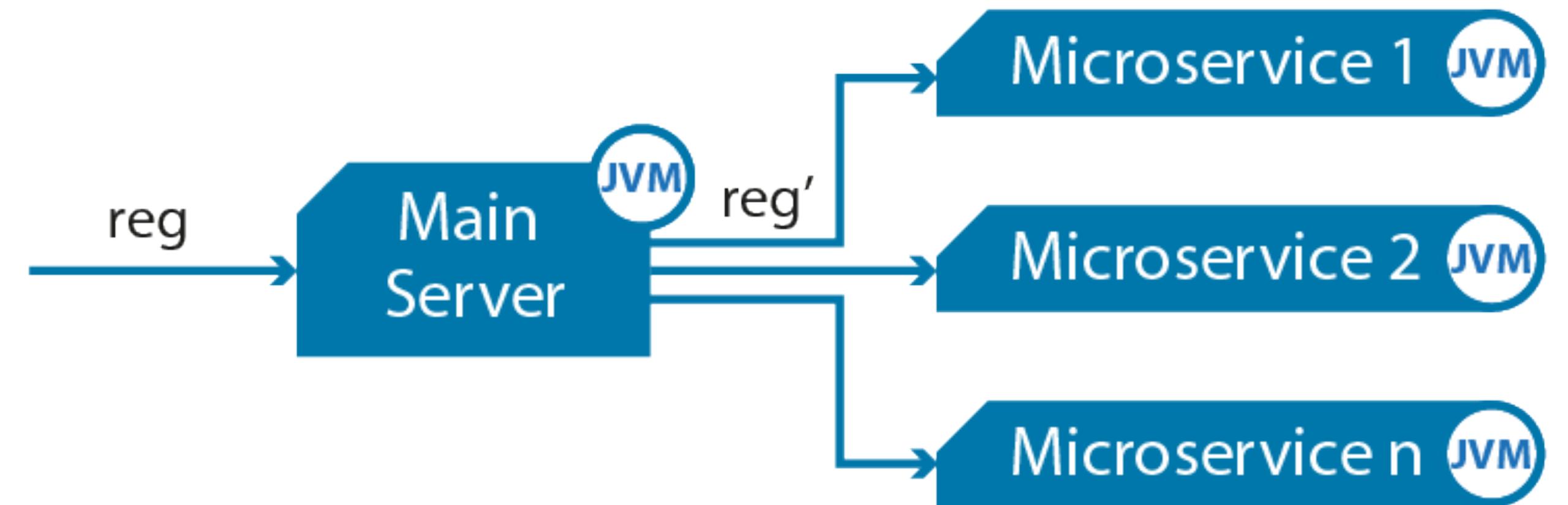


```
<profiles>
  <profile>
    <!-- Define a profile for releases of the application -->
    <!-- mvn release:prepare -P release -DskipTests=true -->
    <!-- mvn release:perform -P release -DskipTests=true -->
    <id>release</id>
    <build>
      <plugins>
        [...]
      </plugins>
    </build>
  </profile>
</profiles>
```

<https://github.com/STAMP-project/botsing/blob/master/pom.xml>

# Variability in software

- Bundles
- Command line parameters
- Plugins
- Configuration files
- *Microservices*

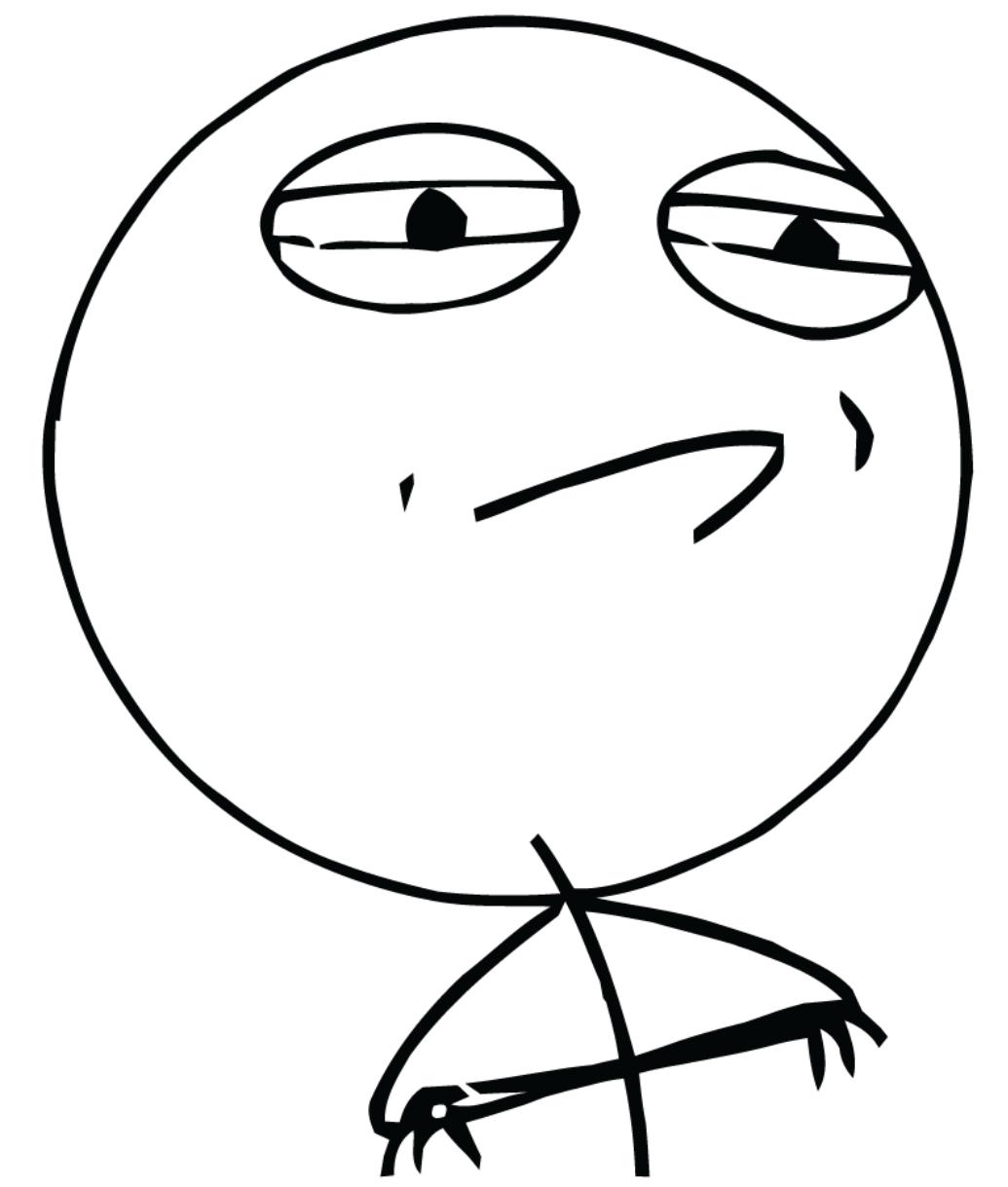


**software variability** is the ability of a software system or artefact to be efficiently extended, changed, **customised** or configured for use in a **particular context**.

# **Does your application have variability?**

Yes, most probably

- **What kind of variability?**
  - Hardware
  - Platform
  - Software
    - Bundles
    - Plugins
    - Command line options
    - Configuration files
    - Microservices
    - ...



# Challenges

<input type="checkbox"/> Option 1	<input type="checkbox"/> Option 2	<input type="checkbox"/> Option 3	<input type="checkbox"/> Option 4
<input type="checkbox"/> Option 5	<input type="checkbox"/> Option 6	<input checked="" type="checkbox"/> Option 7	<input checked="" type="checkbox"/> Option 8
<input checked="" type="checkbox"/> Option 9	<input checked="" type="checkbox"/> Option 10	<input type="checkbox"/> Option 11	<input type="checkbox"/> Option 12
<input type="checkbox"/> Option 13	<input type="checkbox"/> Option 14	<input checked="" type="checkbox"/> Option 15	<input type="checkbox"/> Option 16
<input type="checkbox"/> Option 17	<input type="checkbox"/> Option 18	<input type="checkbox"/> Option 19	<input type="checkbox"/> Option 20
<input checked="" type="checkbox"/> Option 21	<input type="checkbox"/> Option 22	<input type="checkbox"/> Option 23	<input checked="" type="checkbox"/> Option 24
<input type="checkbox"/> Option 25	<input type="checkbox"/> Option 26	<input type="checkbox"/> Option 27	<input type="checkbox"/> Option 28
<input type="checkbox"/> Option 29	<input checked="" type="checkbox"/> Option 30	<input type="checkbox"/> Option 31	<input type="checkbox"/> Option 32
<input type="checkbox"/> Option 33			



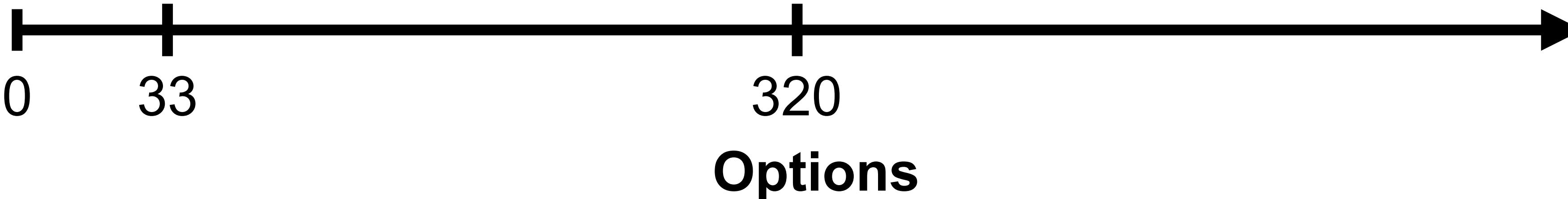
A unique product for every person on this planet  
(8,589,934,592 products)

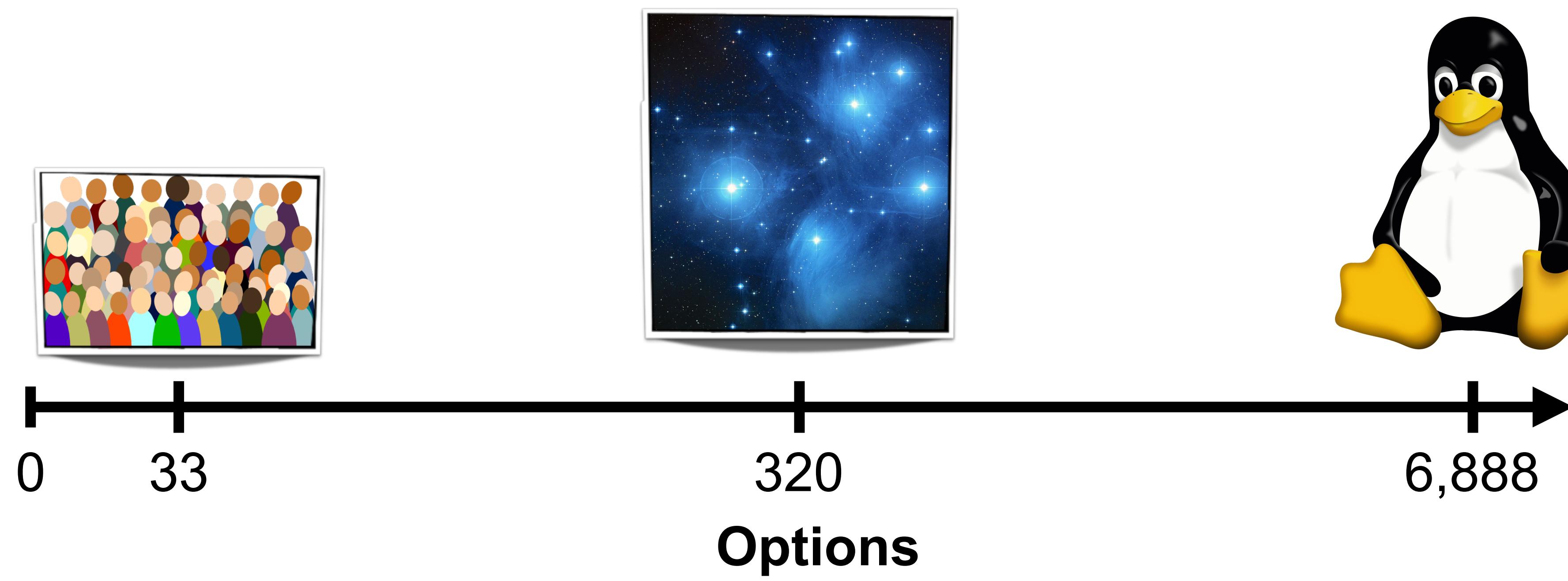
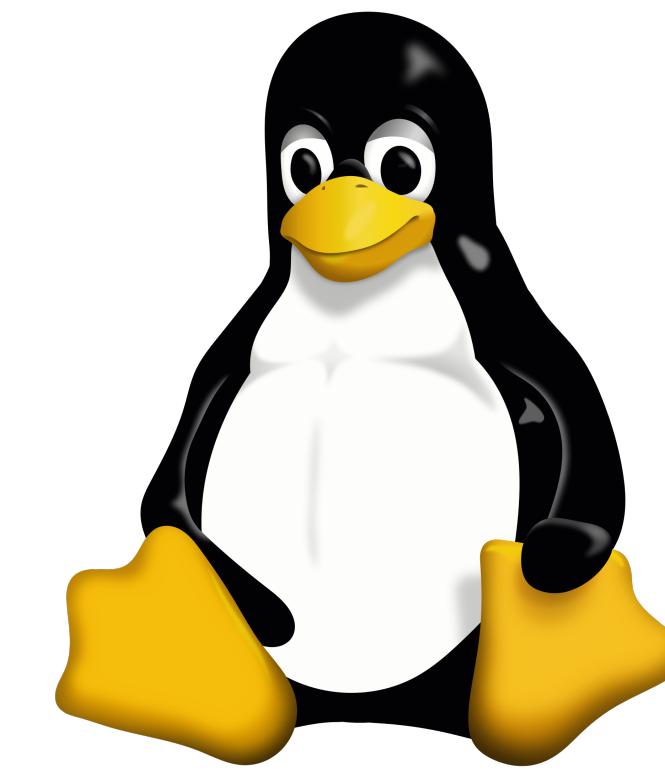


<input type="checkbox"/> Option 1	<input type="checkbox"/> Option 2	<input type="checkbox"/> Option 3	<input type="checkbox"/> Option 4	<input type="checkbox"/> Option 5	<input type="checkbox"/> Option 6	<input type="checkbox"/> Option 7	<input type="checkbox"/> Option 8	<input type="checkbox"/> Option 9	<input type="checkbox"/> Option 10	<input type="checkbox"/> Option 11	<input type="checkbox"/> Option 12	<input type="checkbox"/> Option 13	<input type="checkbox"/> Option 14	<input type="checkbox"/> Option 15	<input type="checkbox"/> Option 16
<input type="checkbox"/> Option 17	<input type="checkbox"/> Option 18	<input type="checkbox"/> Option 19	<input type="checkbox"/> Option 20	<input type="checkbox"/> Option 21	<input type="checkbox"/> Option 22	<input type="checkbox"/> Option 23	<input type="checkbox"/> Option 24	<input type="checkbox"/> Option 25	<input type="checkbox"/> Option 26	<input type="checkbox"/> Option 27	<input type="checkbox"/> Option 28	<input checked="" type="checkbox"/> Option 29	<input type="checkbox"/> Option 30	<input type="checkbox"/> Option 31	<input type="checkbox"/> Option 32
<input checked="" type="checkbox"/> Option 33	<input type="checkbox"/> Option 34	<input type="checkbox"/> Option 35	<input type="checkbox"/> Option 36	<input type="checkbox"/> Option 37	<input type="checkbox"/> Option 38	<input type="checkbox"/> Option 39	<input type="checkbox"/> Option 40	<input type="checkbox"/> Option 41	<input checked="" type="checkbox"/> Option 42	<input checked="" type="checkbox"/> Option 43	<input type="checkbox"/> Option 44	<input type="checkbox"/> Option 45	<input type="checkbox"/> Option 46	<input type="checkbox"/> Option 47	<input type="checkbox"/> Option 48
<input type="checkbox"/> Option 49	<input type="checkbox"/> Option 50	<input checked="" type="checkbox"/> Option 51	<input type="checkbox"/> Option 52	<input type="checkbox"/> Option 53	<input type="checkbox"/> Option 54	<input type="checkbox"/> Option 55	<input type="checkbox"/> Option 56	<input checked="" type="checkbox"/> Option 57	<input type="checkbox"/> Option 58	<input type="checkbox"/> Option 59	<input checked="" type="checkbox"/> Option 60	<input type="checkbox"/> Option 61	<input checked="" type="checkbox"/> Option 62	<input type="checkbox"/> Option 63	<input type="checkbox"/> Option 64
<input type="checkbox"/> Option 65	<input type="checkbox"/> Option 66	<input type="checkbox"/> Option 67	<input type="checkbox"/> Option 68	<input type="checkbox"/> Option 69	<input type="checkbox"/> Option 70	<input type="checkbox"/> Option 71	<input type="checkbox"/> Option 72	<input type="checkbox"/> Option 73	<input checked="" type="checkbox"/> Option 74	<input type="checkbox"/> Option 75	<input type="checkbox"/> Option 76	<input checked="" type="checkbox"/> Option 77	<input type="checkbox"/> Option 78	<input checked="" type="checkbox"/> Option 79	<input checked="" type="checkbox"/> Option 80
<input type="checkbox"/> Option 81	<input type="checkbox"/> Option 82	<input type="checkbox"/> Option 83	<input type="checkbox"/> Option 84	<input type="checkbox"/> Option 85	<input type="checkbox"/> Option 86	<input checked="" type="checkbox"/> Option 87	<input type="checkbox"/> Option 88	<input type="checkbox"/> Option 89	<input type="checkbox"/> Option 90	<input type="checkbox"/> Option 91	<input type="checkbox"/> Option 92	<input type="checkbox"/> Option 93	<input type="checkbox"/> Option 94	<input type="checkbox"/> Option 95	<input type="checkbox"/> Option 96
<input type="checkbox"/> Option 97	<input checked="" type="checkbox"/> Option 98	<input type="checkbox"/> Option 99	<input checked="" type="checkbox"/> Option 100	<input type="checkbox"/> Option 101	<input type="checkbox"/> Option 102	<input type="checkbox"/> Option 103	<input checked="" type="checkbox"/> Option 104	<input type="checkbox"/> Option 105	<input type="checkbox"/> Option 106	<input type="checkbox"/> Option 107	<input type="checkbox"/> Option 108	<input type="checkbox"/> Option 109	<input type="checkbox"/> Option 110	<input type="checkbox"/> Option 111	<input type="checkbox"/> Option 112
<input checked="" type="checkbox"/> Option 113	<input type="checkbox"/> Option 114	<input type="checkbox"/> Option 115	<input type="checkbox"/> Option 116	<input type="checkbox"/> Option 117	<input type="checkbox"/> Option 118	<input type="checkbox"/> Option 119	<input type="checkbox"/> Option 120	<input type="checkbox"/> Option 121	<input type="checkbox"/> Option 122	<input type="checkbox"/> Option 123	<input type="checkbox"/> Option 124	<input type="checkbox"/> Option 125	<input type="checkbox"/> Option 126	<input checked="" type="checkbox"/> Option 127	<input type="checkbox"/> Option 128
<input type="checkbox"/> Option 129	<input type="checkbox"/> Option 130	<input type="checkbox"/> Option 131	<input type="checkbox"/> Option 132	<input type="checkbox"/> Option 133	<input type="checkbox"/> Option 134	<input type="checkbox"/> Option 135	<input type="checkbox"/> Option 136	<input type="checkbox"/> Option 137	<input type="checkbox"/> Option 138	<input type="checkbox"/> Option 139	<input type="checkbox"/> Option 140	<input type="checkbox"/> Option 141	<input type="checkbox"/> Option 142	<input type="checkbox"/> Option 143	<input type="checkbox"/> Option 144
<input type="checkbox"/> Option 145	<input type="checkbox"/> Option 146	<input type="checkbox"/> Option 147	<input type="checkbox"/> Option 148	<input type="checkbox"/> Option 149	<input type="checkbox"/> Option 150	<input type="checkbox"/> Option 151	<input type="checkbox"/> Option 152	<input type="checkbox"/> Option 153	<input type="checkbox"/> Option 154	<input checked="" type="checkbox"/> Option 155	<input type="checkbox"/> Option 156	<input type="checkbox"/> Option 157	<input checked="" type="checkbox"/> Option 158	<input type="checkbox"/> Option 159	<input checked="" type="checkbox"/> Option 160
<input type="checkbox"/> Option 161	<input type="checkbox"/> Option 162	<input type="checkbox"/> Option 163	<input type="checkbox"/> Option 164	<input type="checkbox"/> Option 165	<input type="checkbox"/> Option 166	<input type="checkbox"/> Option 167	<input type="checkbox"/> Option 168	<input type="checkbox"/> Option 169	<input type="checkbox"/> Option 170	<input type="checkbox"/> Option 171	<input checked="" type="checkbox"/> Option 172	<input checked="" type="checkbox"/> Option 173	<input type="checkbox"/> Option 174	<input type="checkbox"/> Option 175	<input type="checkbox"/> Option 176
<input checked="" type="checkbox"/> Option 177	<input checked="" type="checkbox"/> Option 178	<input type="checkbox"/> Option 179	<input type="checkbox"/> Option 180	<input type="checkbox"/> Option 181	<input type="checkbox"/> Option 182	<input type="checkbox"/> Option 183	<input type="checkbox"/> Option 184	<input checked="" type="checkbox"/> Option 185	<input type="checkbox"/> Option 186	<input checked="" type="checkbox"/> Option 187	<input type="checkbox"/> Option 188	<input type="checkbox"/> Option 189	<input type="checkbox"/> Option 190	<input checked="" type="checkbox"/> Option 191	<input type="checkbox"/> Option 192
<input type="checkbox"/> Option 193	<input type="checkbox"/> Option 194	<input checked="" type="checkbox"/> Option 195	<input checked="" type="checkbox"/> Option 196	<input type="checkbox"/> Option 197	<input type="checkbox"/> Option 198	<input type="checkbox"/> Option 199	<input type="checkbox"/> Option 200	<input type="checkbox"/> Option 201	<input type="checkbox"/> Option 202	<input type="checkbox"/> Option 203	<input type="checkbox"/> Option 204	<input type="checkbox"/> Option 205	<input checked="" type="checkbox"/> Option 206	<input type="checkbox"/> Option 207	<input type="checkbox"/> Option 208
<input type="checkbox"/> Option 209	<input type="checkbox"/> Option 210	<input type="checkbox"/> Option 211	<input type="checkbox"/> Option 212	<input type="checkbox"/> Option 213	<input type="checkbox"/> Option 214	<input type="checkbox"/> Option 215	<input type="checkbox"/> Option 216	<input type="checkbox"/> Option 217	<input type="checkbox"/> Option 218	<input type="checkbox"/> Option 219	<input type="checkbox"/> Option 220	<input type="checkbox"/> Option 221	<input type="checkbox"/> Option 222	<input type="checkbox"/> Option 223	<input type="checkbox"/> Option 224
<input type="checkbox"/> Option 225	<input type="checkbox"/> Option 226	<input type="checkbox"/> Option 227	<input type="checkbox"/> Option 228	<input type="checkbox"/> Option 229	<input type="checkbox"/> Option 230	<input type="checkbox"/> Option 231	<input type="checkbox"/> Option 232	<input checked="" type="checkbox"/> Option 233	<input type="checkbox"/> Option 234	<input type="checkbox"/> Option 235	<input type="checkbox"/> Option 236	<input type="checkbox"/> Option 237	<input type="checkbox"/> Option 238	<input checked="" type="checkbox"/> Option 239	<input type="checkbox"/> Option 240
<input type="checkbox"/> Option 241	<input checked="" type="checkbox"/> Option 242	<input type="checkbox"/> Option 243	<input checked="" type="checkbox"/> Option 244	<input type="checkbox"/> Option 245	<input type="checkbox"/> Option 246	<input checked="" type="checkbox"/> Option 247	<input type="checkbox"/> Option 248	<input type="checkbox"/> Option 249	<input type="checkbox"/> Option 250	<input type="checkbox"/> Option 251	<input type="checkbox"/> Option 252	<input type="checkbox"/> Option 253	<input type="checkbox"/> Option 254	<input type="checkbox"/> Option 255	<input type="checkbox"/> Option 256
<input checked="" type="checkbox"/> Option 257	<input type="checkbox"/> Option 258	<input type="checkbox"/> Option 259	<input type="checkbox"/> Option 260	<input checked="" type="checkbox"/> Option 261	<input type="checkbox"/> Option 262	<input type="checkbox"/> Option 263	<input type="checkbox"/> Option 264	<input type="checkbox"/> Option 265	<input type="checkbox"/> Option 266	<input checked="" type="checkbox"/> Option 267	<input type="checkbox"/> Option 268	<input type="checkbox"/> Option 269	<input checked="" type="checkbox"/> Option 270	<input checked="" type="checkbox"/> Option 271	<input type="checkbox"/> Option 272
<input type="checkbox"/> Option 273	<input type="checkbox"/> Option 274	<input type="checkbox"/> Option 275	<input type="checkbox"/> Option 276	<input type="checkbox"/> Option 277	<input type="checkbox"/> Option 278	<input type="checkbox"/> Option 279	<input type="checkbox"/> Option 280	<input checked="" type="checkbox"/> Option 281	<input type="checkbox"/> Option 282	<input type="checkbox"/> Option 283	<input checked="" type="checkbox"/> Option 284	<input type="checkbox"/> Option 285	<input type="checkbox"/> Option 286	<input type="checkbox"/> Option 287	<input checked="" type="checkbox"/> Option 288
<input type="checkbox"/> Option 289	<input type="checkbox"/> Option 290	<input type="checkbox"/> Option 291	<input type="checkbox"/> Option 292	<input type="checkbox"/> Option 293	<input type="checkbox"/> Option 294	<input type="checkbox"/> Option 295	<input type="checkbox"/> Option 296	<input type="checkbox"/> Option 297	<input type="checkbox"/> Option 298	<input type="checkbox"/> Option 299	<input type="checkbox"/> Option 300	<input type="checkbox"/> Option 301	<input type="checkbox"/> Option 302	<input type="checkbox"/> Option 303	<input type="checkbox"/> Option 304
<input type="checkbox"/> Option 305	<input type="checkbox"/> Option 306	<input type="checkbox"/> Option 307	<input type="checkbox"/> Option 308	<input type="checkbox"/> Option 309	<input type="checkbox"/> Option 310	<input type="checkbox"/> Option 311	<input type="checkbox"/> Option 312	<input type="checkbox"/> Option 313	<input type="checkbox"/> Option 314	<input type="checkbox"/> Option 315	<input type="checkbox"/> Option 316	<input type="checkbox"/> Option 317	<input type="checkbox"/> Option 318	<input type="checkbox"/> Option 319	<input type="checkbox"/> Option 320



More products than the  
estimated atoms in the universe  
( $2.135987 \times 10^{96}$  products)





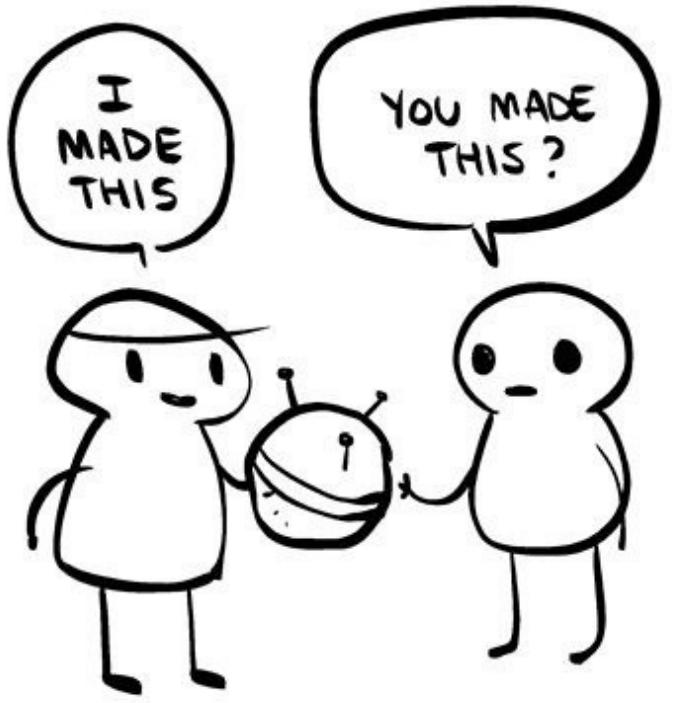
**How to...**

**...define requirements?**

**...develop?**

**...test?**

**...maintain?**



# Clone and Own

(Variability 101)

zeit / next.js

Used by ▾

64k

Watch ▾

852

Star

45.1k

Fork

6.4k

numpy / numpy

Sponsor

Used by ▾

318k

Watch ▾

485

Star

12.9k

Fork

4.3k

ansible / ansible

Used by ▾

10.2k

Watch ▾

2k

Star

41.8k

Fork

18.4k

ethereum / solidity

Watch ▾

493

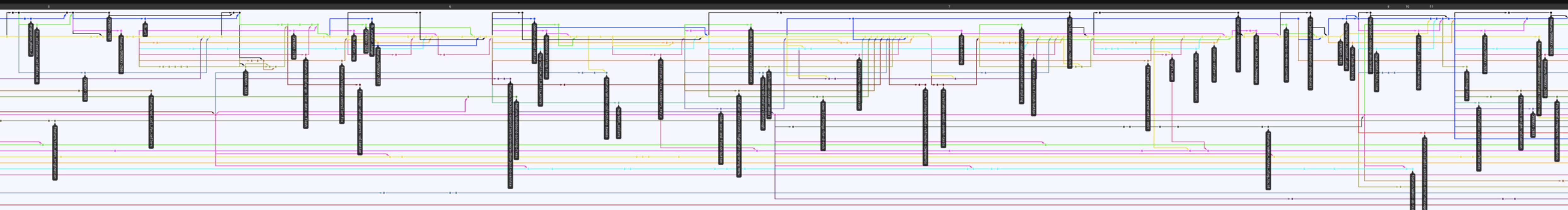
Star

7.7k

Fork

2.1k

**Problem: how to manage the evolution of a large number of copies?**



# Hudson



Hudson - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://kohsuke.sfbay/hudson/ Go

## Hudson

Hudson

New Job

Configure

Reload Config

Build Queue

hudson	Idle
jaxb-ri	Idle

Build Executor Status

No.	Status
1	Idle
2	Idle
3	Building javanet-maven-repository-daemon #826
4	Building jaxb-ri #3181
5	Building glassfish #105
6	Idle

All JAX-WS JAXB Tango java.net +

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
dtd-parser	6 months (#8)	N/A	1 minute
fi	28 days (#586)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
istack-commons	12 days (#19)	16 days (#5)	14 seconds
japex	3 days (#55)	9 hours (#64)	1 minute
java-ws-xml community discussion updater	4 minutes (#16146)	10 hours (#16125)	1 minute
java.net acl processor	18 hours (#162)	N/A	0 seconds

ENABLE AUTO REFRESH

Internet



# Jenkins

Jenkins

search ? Claudia Meindl | log out

Back to Project

Status

Changes

Console Output

View as plain text

View Build Information

Git Build Data

Previous Build

### Console Output

```
Skipping 1.196 KB.. Full Log
KoEttlJQNjBwdnF2sVayAwD4vuoDpQAAAA==[0ms skipping: [teamwiesn.com]

TASK [smtp-server : include] ****
skipping: [teamwiesn.com]

TASK [ssl : include] ****
included: /var/lib/jenkins/jobs/Rollout-system/workspace/roles/ssl/tasks/setup.yml for
teamwiesn.com

TASK [ssl : install ssl packages] ****
ok: [teamwiesn.com] => (item=[u'ssl-cert', u'sslscan'])

TASK [ssl : Install ssl certificates] ****
ok: [teamwiesn.com] => (item=teamwiesn.crt)

TASK [ssl : Install ssl keys] ****
ok: [teamwiesn.com] => (item=teamwiesn.key)

TASK [ssl : Generate dhparam key] ****
ok: [teamwiesn.com]

TASK [ssl : Check ssl forward secrecy key permission] ****
ok: [teamwiesn.com]

TASK [ssl : include] ****
skipping: [teamwiesn.com]

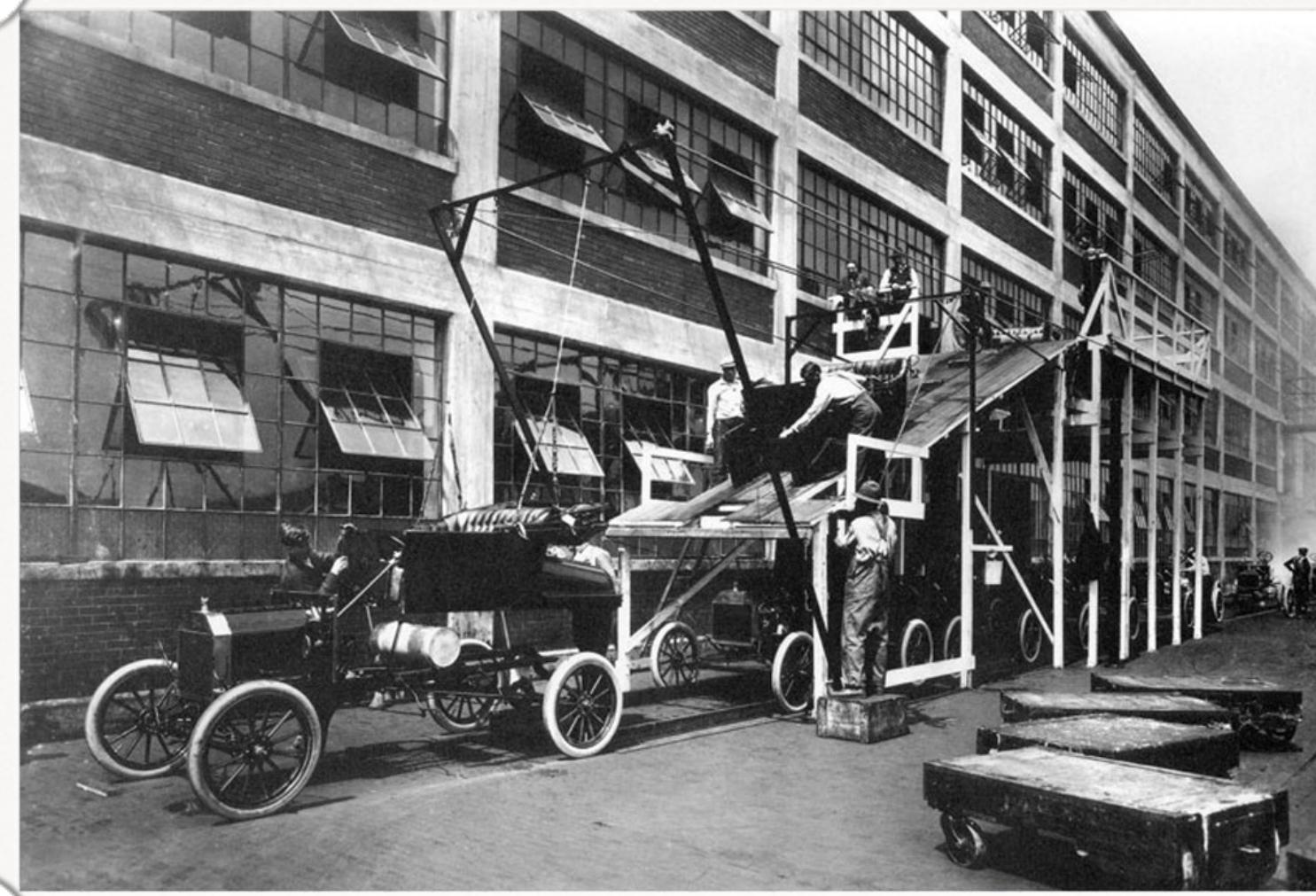
TASK [pip : Check to see if pip is already installed.] ****
ok: [teamwiesn.com]
```

- **Fast**
- **Cheap (at first)**
- **Limited**
  - Manage (large number of) copies
  - Maintenance cost at each evolution
  - Inconsistent evolution of copies



# Software Product Line

(Variability level over 9000!)



## Henry Ford 1901



**Volkswagen**

**Models**

## Build your Volkswagen

Locate me Load Filter your Search

Golf From \$19,595<sup>1</sup>  
Golf GTI From \$30,595<sup>1</sup>  
Golf R From \$42,065<sup>1</sup>  
e-Golf From \$36,355<sup>1</sup>  
Golf SportWagen From \$24,195<sup>1</sup>  
Golf Alltrack From \$34,345<sup>1</sup>  
Beetle From \$21,895<sup>1</sup>  
Beetle Convertible From \$25,895<sup>1</sup>  
Jetta From \$20,995<sup>1</sup>  
New Tiguan From \$27,995<sup>1</sup>  
Passat From \$35,690<sup>1</sup>  
Atlas From \$35,690<sup>1</sup>

**Build and Price**

**Special Offers**

**Used Vehicles**

**VW Owners**

**VW Finance**

**More**

EN | FR

**Golf GTI**

Exterior colour (1/3)

Summary

Gallery 360° View Virtual Reality

Choose your exterior colour

Top coat - metallic

Dark Iron Blue Metallic No additional costs  
Platinum Grey Metallic No additional costs

Go to options

**Golf GTI**

From \$30,595<sup>1</sup>

**Key Features**

- 17" Brooklyn alloy wheels with all-season tires
- Ambient interior lighting
- "App-Connect" smartphone integration (Android Auto, Apple Carplay, and Mirrorlink)
- Leather multi-function sport steering wheel

My standard equipment

Compare trims

Choose your engine and transmission

2.0 TSI 6 speed manual From \$30,595<sup>1</sup>

2.0 TSI 6 speed automatic with Tiptronic® From \$31,995<sup>1</sup>

Build your Volkswagen

Load Load from My Garage Load a previous configuration

**Compare Golf GTI Trims**

**Standard Equipment**

Compare standard equipment on two trims to find the car that fits you.

Show only differences.

	GTI	Autobahn
17" Brooklyn alloy wheels with all-season tires	●	—
18" Milton alloy wheels with all-season tires	○	●
8 speakers	●	—
Fender® Premium Audio System	○	●
Cloth seating surfaces	●	○
Leather seating surfaces	○	●
Front fog lights	●	—
Front fog lights with static cornering lights	○	●
LED daytime running lights	●	—

**Volkswagen**

**Models**

**Build and Price**

**Special Offers**

**Used Vehicles**

**VW Owners**

**VW Finance**

**More**

EN | FR

Go to Home Page

Notifications 1 Logged in as Davis

## Choose a product

All categories > Clothing / Apparel > Women > Tank-tops

All Alternative American Apparel Anvil Bella + Canvas Next Level

Bella + Canvas 3001  
Unisex Short Sleeve Jersey T-Shirt with Tear Away Label  
Starting from \$13

Anvil 980 Lightweight Fashion Short Sleeve T-Shirt with Tear Away™ Label  
Starting from \$17.25

Bella + Canvas 3413  
Triblend Short Sleeve T-Shirt with Tear Away Label  
Starting from \$13.50

LA Apparel 20001 Fine Jersey Short Sleeve Men T-Shirt w/ Tear Away Label  
\$13.95

Choose between hundreds of product variations



IKEA Home Planner FY08 - 1.9.4

You have chosen:  
W/G1530  
W: 15"  
H: 30 3/8"  
D: 12 1/4"

Handle: No handle  
Decor: No decor  
Frame color: #2583:10156  
Front Color: ABSTRAKT #2583:10156

Base cabinets 12 7/8" deep  
High cabinets  
High cabinets 12 7/8" deep  
Wall cabinets  
30 3/8" high  
Corner cabinets  
Cabinets with doors  
Cabinets with glass door(s)  
Cabinets for microwave or  
Open shelves

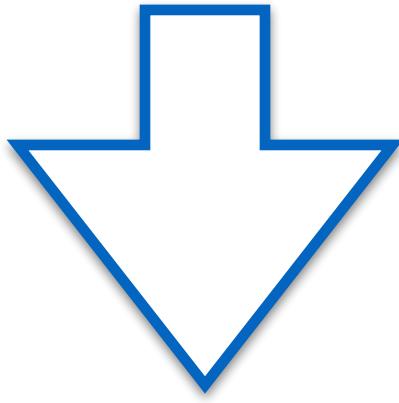
CW2530	W1230	W1530	W1830	W2130	W24301	W2430	W3030
25x25x30"	12x30"	15x30"	18x30"	21x30"	24x30"	24x30"	30x30"

BEN&JERRY'S Products Range

500ml TUBS

LOW FAT FROZEN YOGURT 3% FAT

A product is built by systematically combining  
**commonalities**, common to all products, and  
**variabilities**, specific to some products.



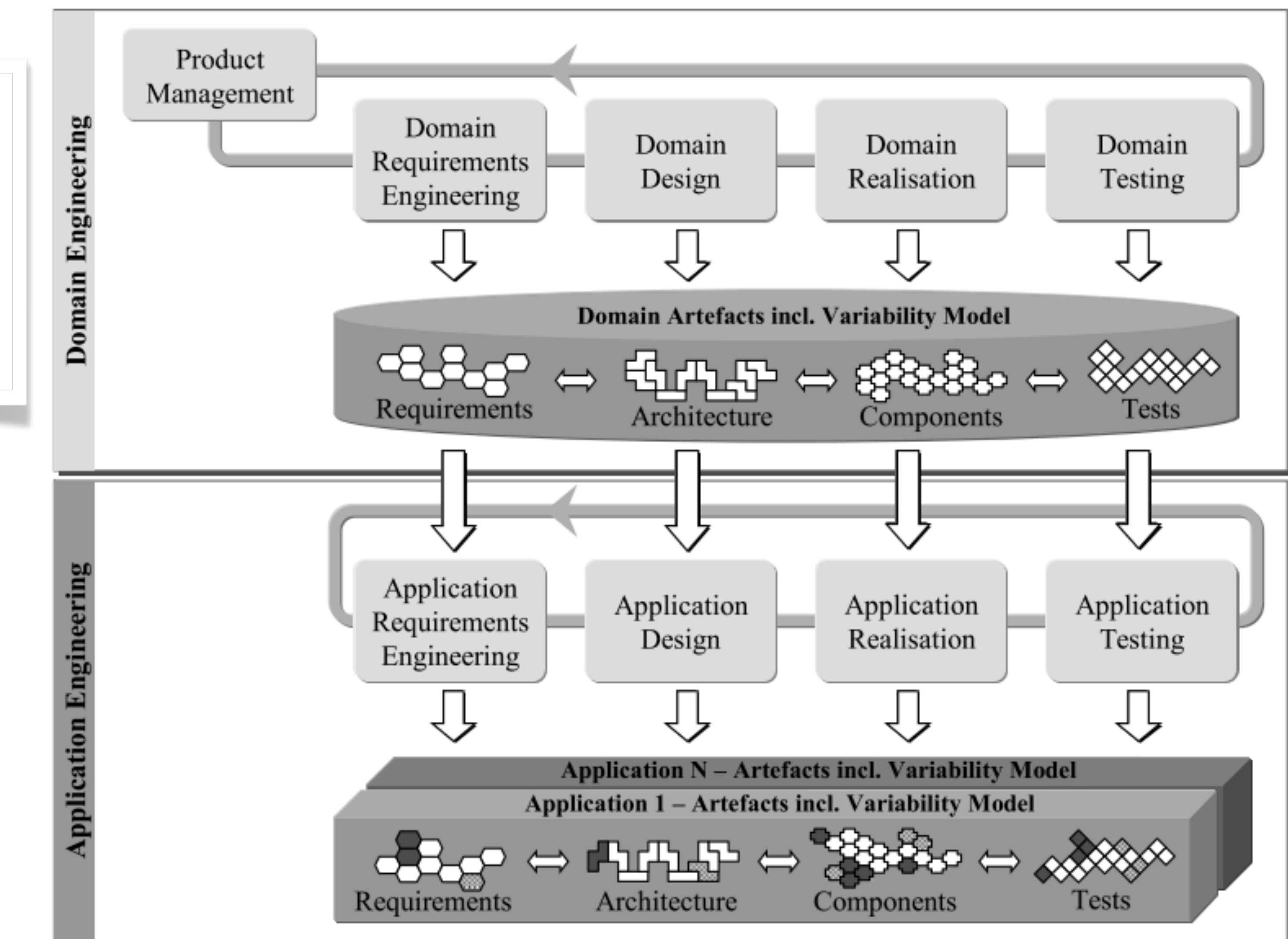
## Features

A **feature** is a characteristic  
or **end-user-visible behaviour**  
of a software system.

# The software product line engineering framework

**Commonalities and variabilities  
definition and realisation**

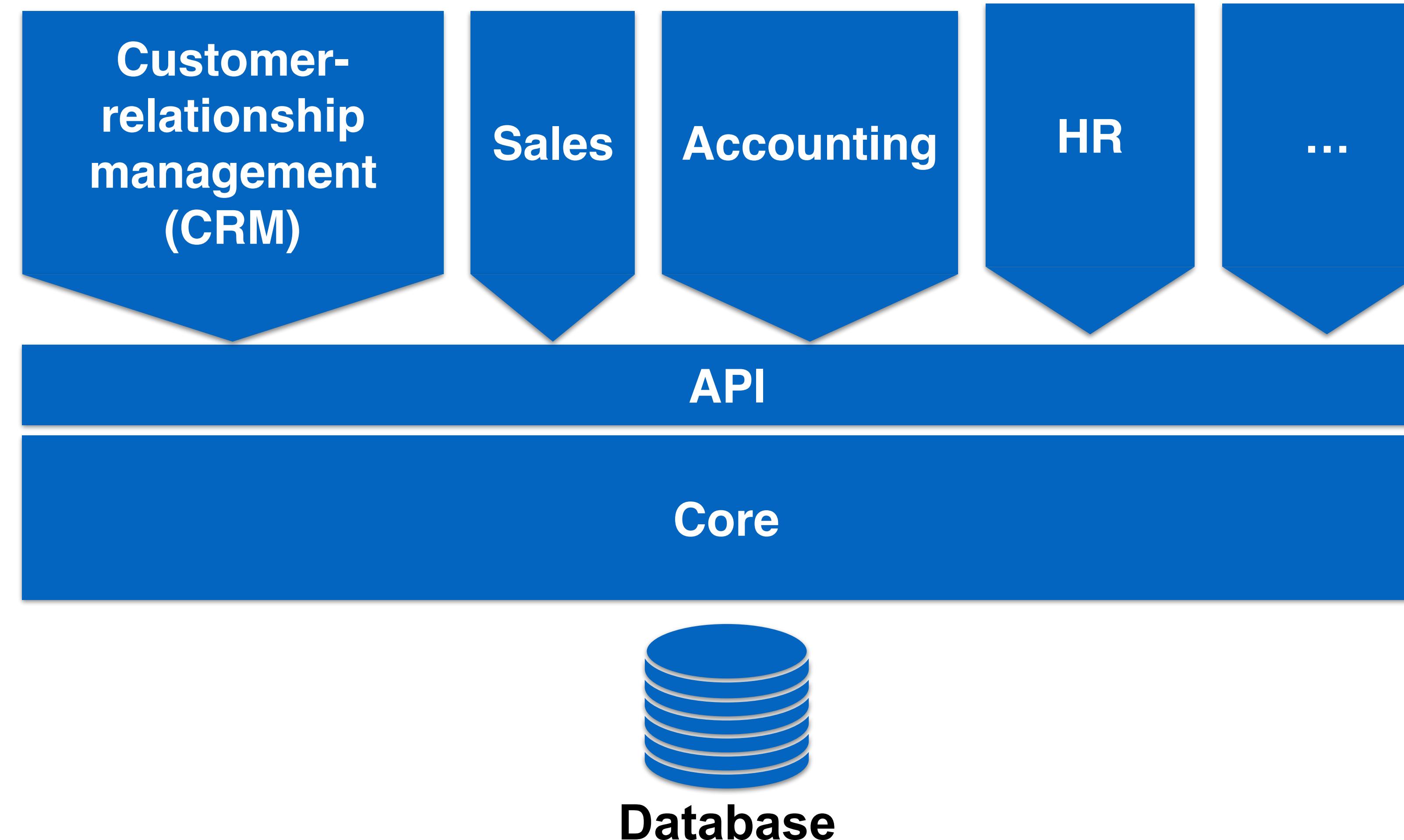
**Product derivation**





**Example: SAP, an Enterprise Resource Planning system (ERP)**

# Integrated management of business processes and operational data

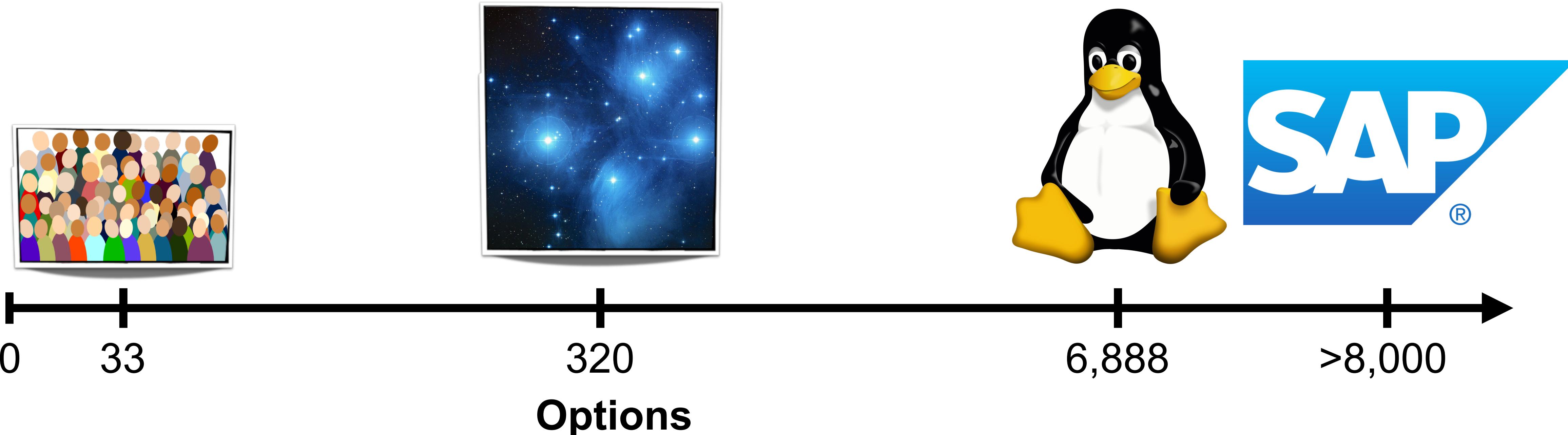


# ERPs in numbers

- SAP was founded in 1972
  - Designed to be extensible from the beginning
- Every year (worldwide)...
  - companies spend \$41B on ERPs
  - SAP alone sells for \$25B
  - SAP systems manage 77% of the world's transaction revenue, and 78% of the world's food

A basic SAP installation has 20,000 database tables,  
**3,000 are configuration tables**

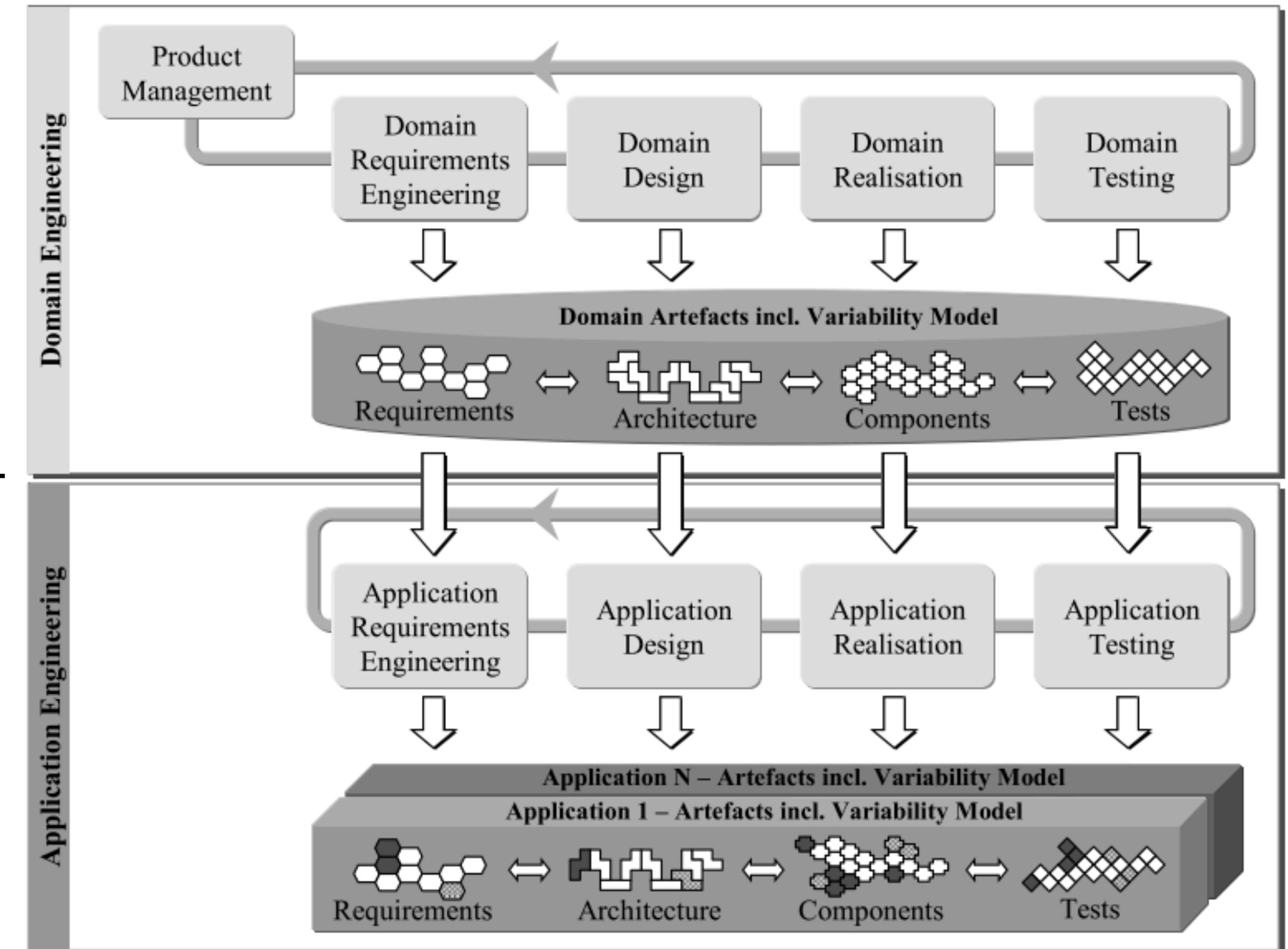
SAP requires ~8,000 configuration decisions at setup  
→ *SAP Configuration Specialist* is an actual job title!



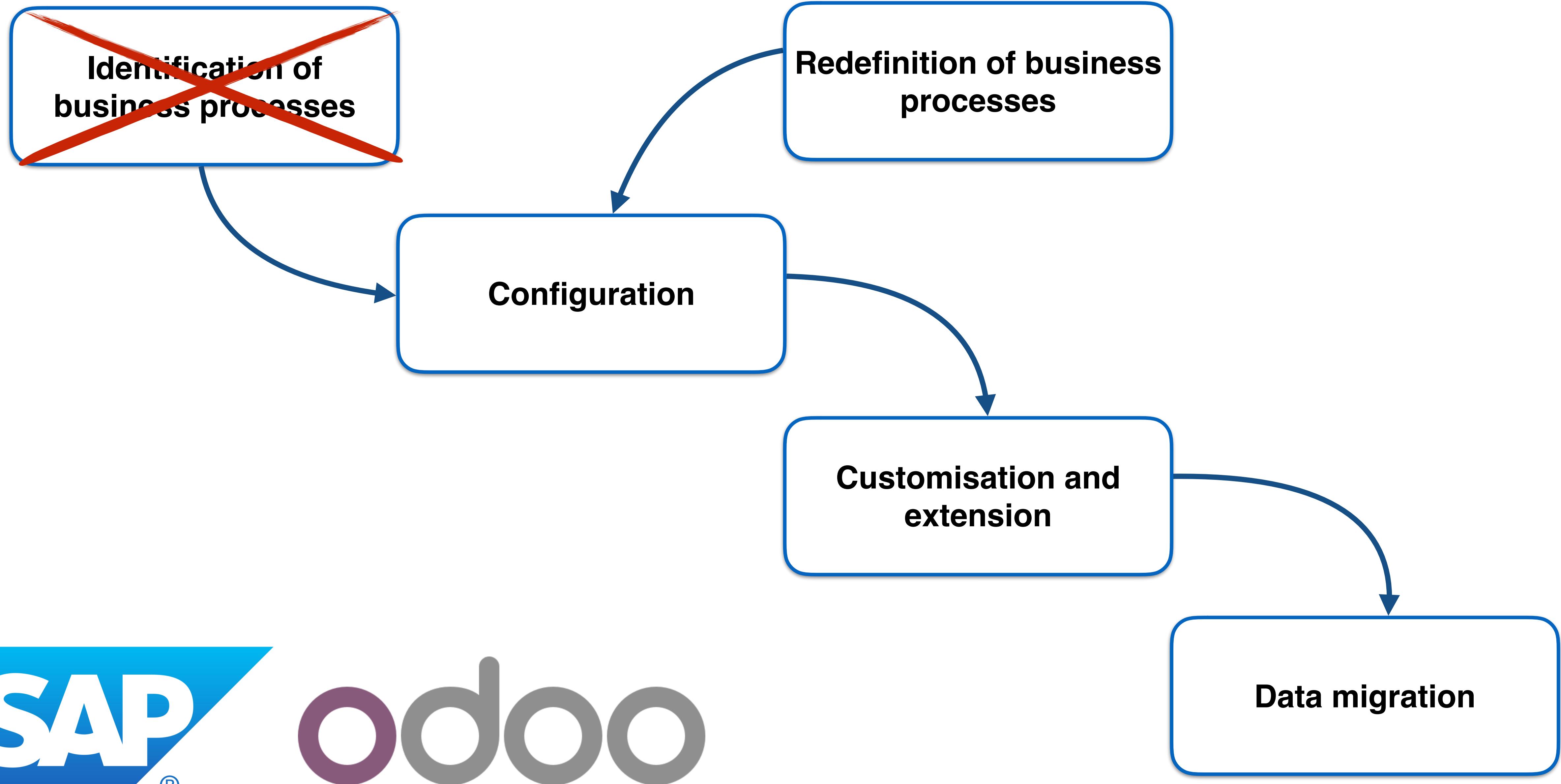
# ERPs in numbers

- Switching to an ERP
  - Costs \$100 million to \$500 million
    - \$30 million in software license fees,
    - \$200 million in consulting fees,
    - millions for hardware,
    - and millions to train managers and employees.
  - Takes from four to six years
- Examples
  - Cisco's ERP implementation took 9 months and \$15 million (best case)
  - Dow Chemical's implementation took 8 years and \$1 billion
  - the U.S. navy spent \$1 billion on four different ERP projects that all failed

- Building the ERP platform
    - Core
    - API
    - Modules
    - Configuration workflow
- 



# Example: Enterprise resource planning systems (ERPs)



# Hall of Fame Software Product Line Conference (SPLC)

- Boeing
- Bosch Group
- CelsiusTech Systems AB
- Cummins, Inc.
- Danfoss
- Ericsson AXE
- FISCAN
- General Motors Powertrain (GMPT)
- Hewlett Packard
- HomeAway
- Lockheed Martin
- LSI Logic
- Lucent
- Market Maker
- Nokia
- Philips (Medical Systems)
- Philips (Software for Television Sets)
- Philips (Telecommunication Switching System)
- Salion, Inc.
- Siemens (syngo.via)
- Toshiba
- U.S. Army Live Traning Transformation
- U.S. Naval Research Laboratory

<http://splc.net/hall-of-fame/>

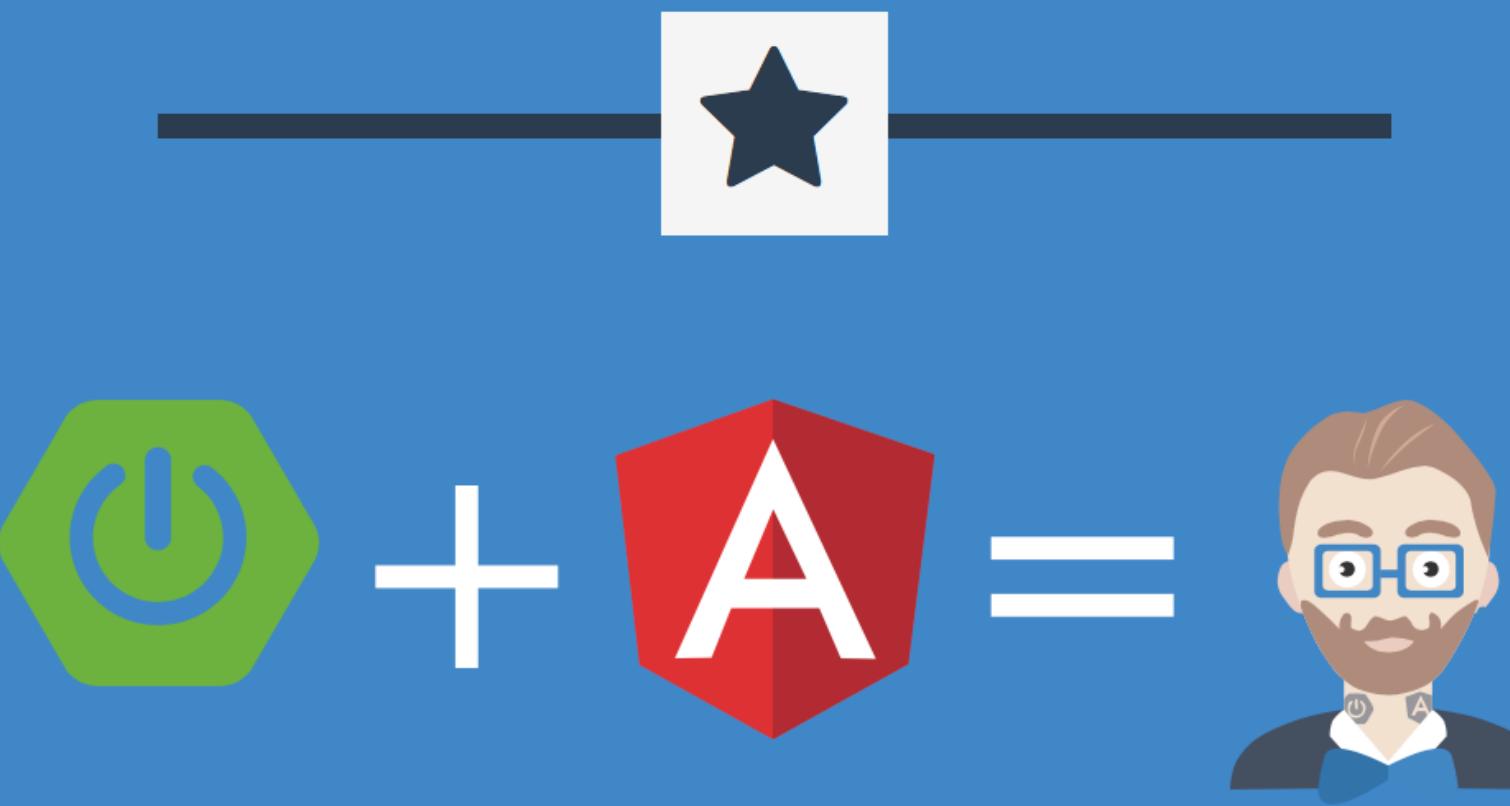
- **Tailor-made**
  - For individual customers
- **Reduced costs**
  - Reusable parts can be combined in different ways
- **Improved quality**
  - Parts can be standardized and checked in isolation and reused and tested in multiple products.
- **Time to market**
  - building on top of existing parts is much faster than developing it entirely from scratch



*Greetings, Java Hipster!*

# Feature Modelling

# What Is JHipster?



JHipster is a development platform to generate, develop and deploy Spring Boot  
+ Angular Web applications and Spring microservices.



<https://start.jhipster.tech/>

**Client Side Options**

- HTML5
- CSS3
- Bootstrap
- AngularJS
- Angular
- jQuery
- Websockets
- Yarn
- Webpack
- Bower
- Gulp
- Sass
- Browsersync
- Karma
- Protractor

**Server Side Options**

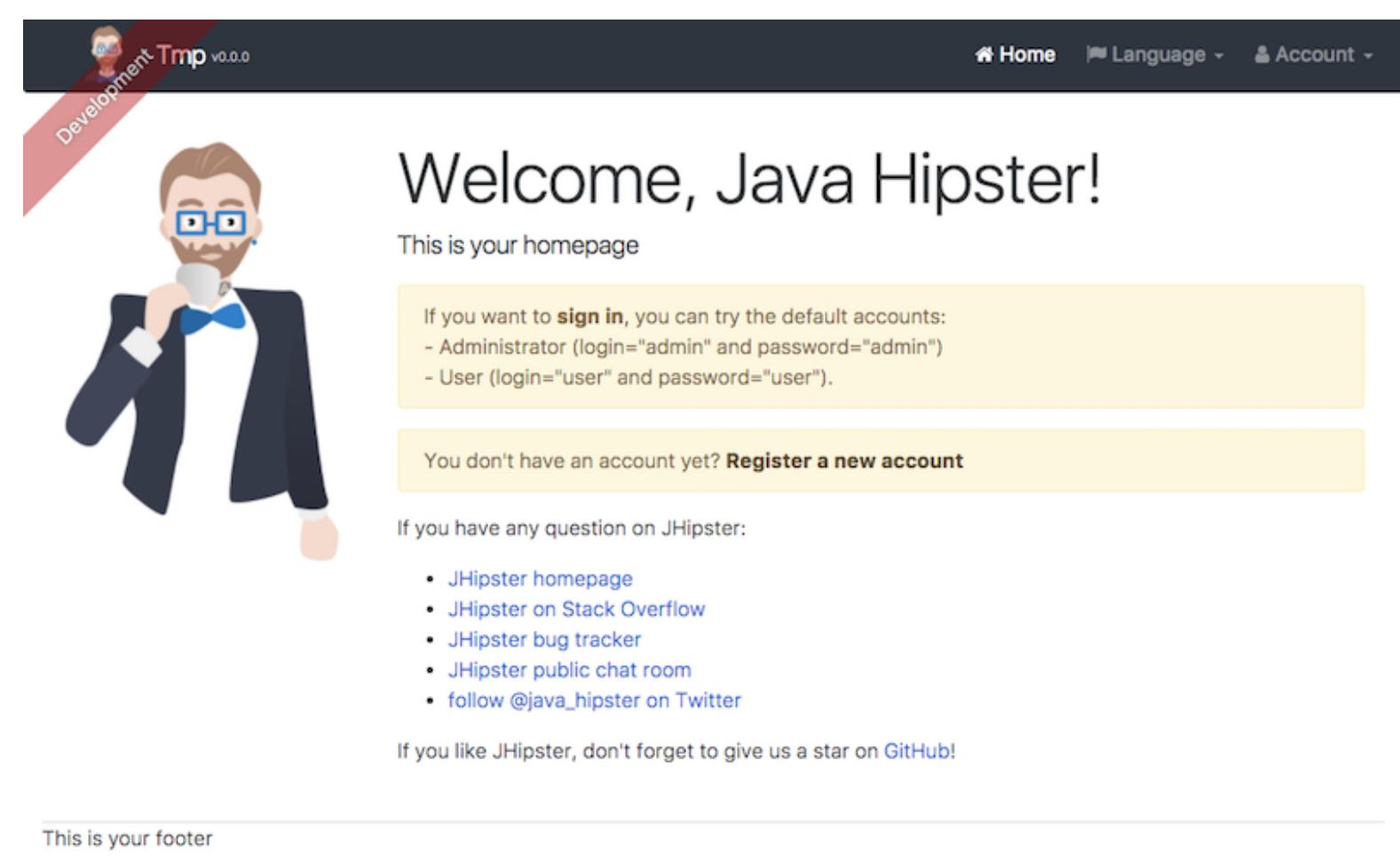
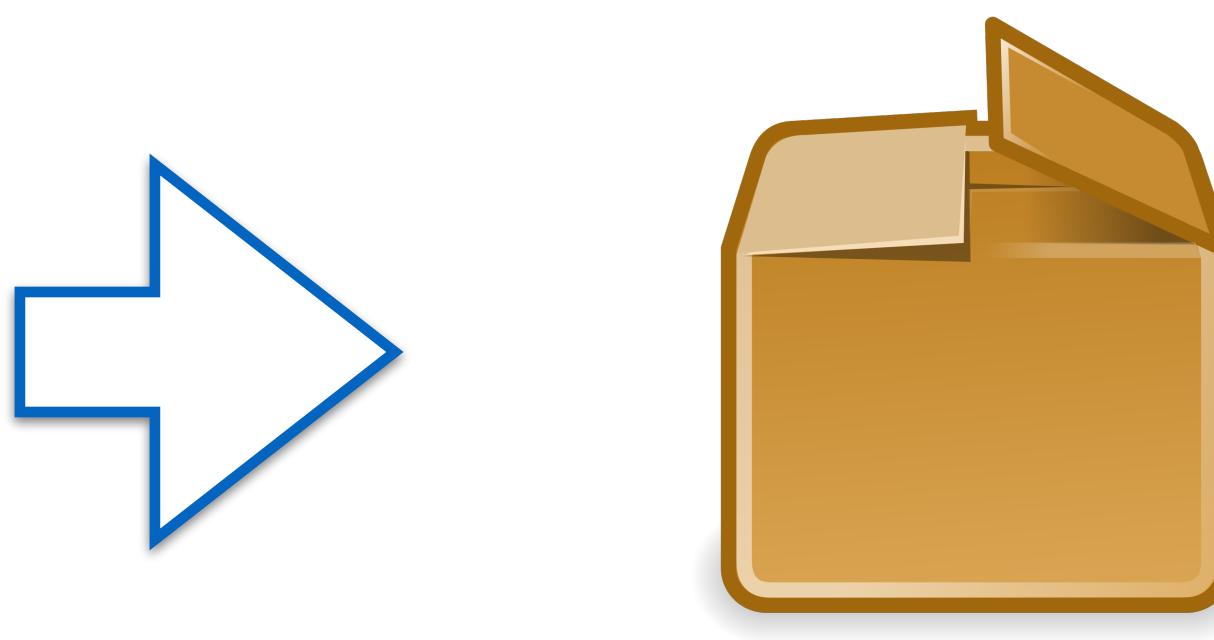
- Spring Boot
- Spring Security
- NETFLIX OSS
- Netflix OSS
- Consul
- Gradle
- Maven
- Hibernate
- Liquibase
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- SQL Server
- MS SQL
- MongoDB
- cassandra
- EhCache
- Hazelcast
- ElasticSearch
- Kafka
- Swagger
- ELK Stack
- ELK Stack
- Prometheus
- Thymeleaf
- Gatling
- Cucumber

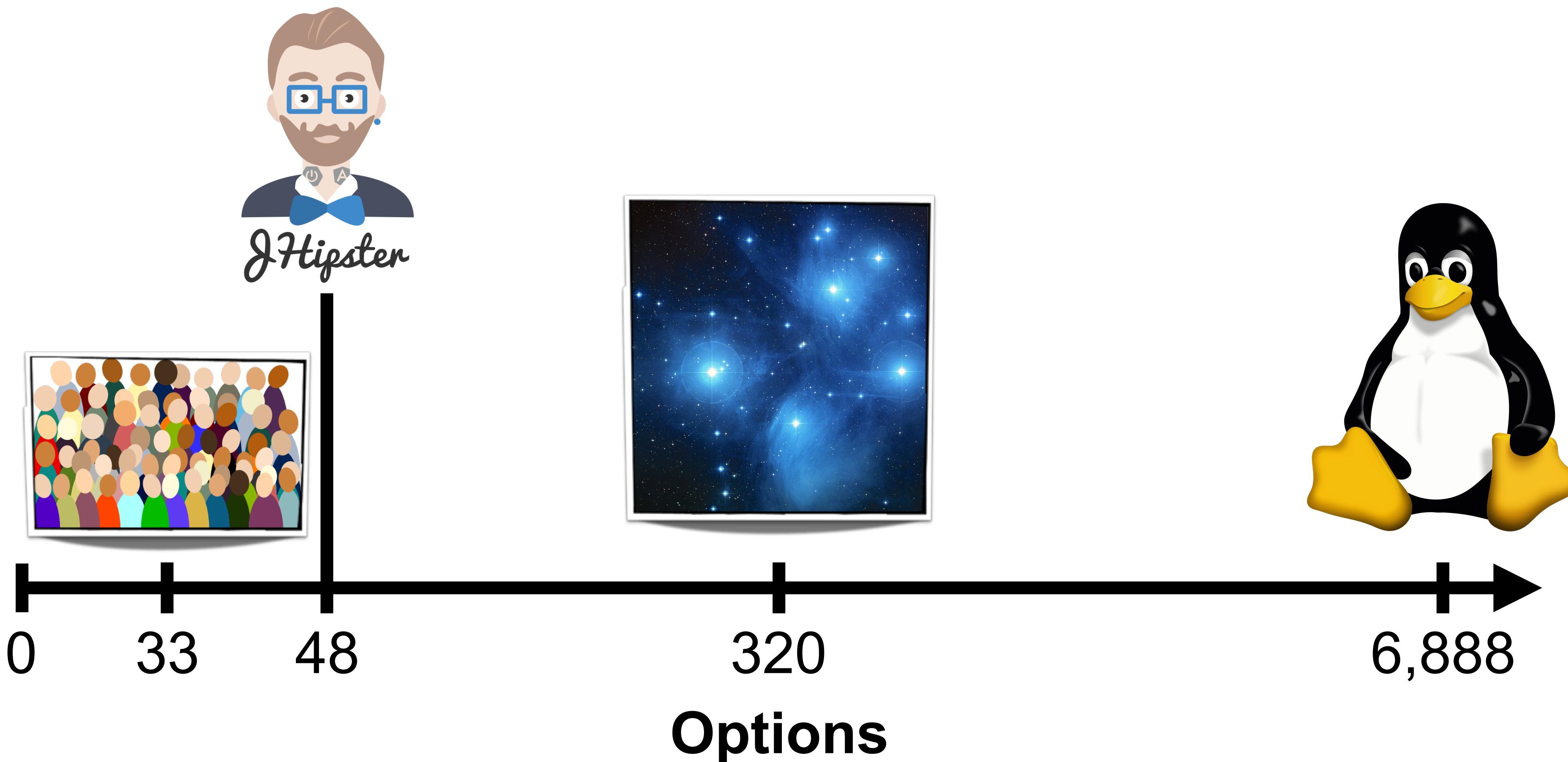
**Deployment Options**

- Docker
- Kubernetes
- Heroku
- Cloud Foundry
- AWS
- Boxfuse
- Rancher
- OpenShift

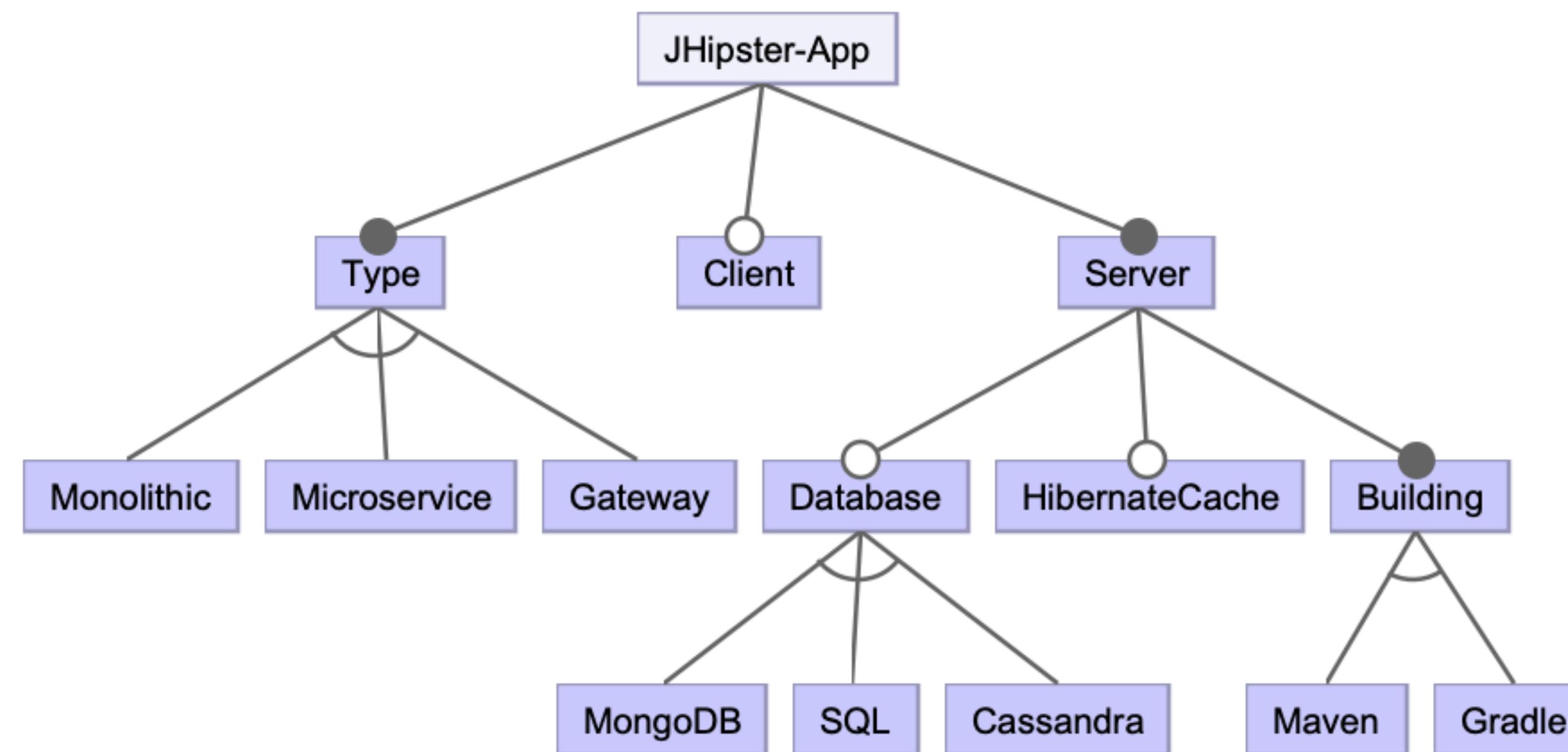
The screenshot shows the NetBeans IDE interface with the title bar "be.unamur.testme.jhipster-0.0.1-SNAPSHOT - NetBeans IDE 8.1". The left sidebar displays the project structure under "Project Navigator" with packages like "be.unamur.info.b314-compiler-0.0.1-SNAPSHOT", "be.unamur.info.methode-progra2-0.0.1", etc., and "Source Packages" containing "be.unamur.testme" and its sub-packages. The main editor window shows the code for "JhipsterApp.java". The code includes annotations for JHipster and Spring profiles, and handles command-line arguments for running the application. A red vertical line highlights a specific section of the code.

```
35 /**
36  * Initializes jhipster.
37  * <p>
38  * Spring profiles can be configured with a program arguments --spring.profiles.active=your-active-profile
39  * <p>
40  * You can find more information on how profiles work with JHipster on <a href="http://jhipster.github.io/profiles/">http://jhipster.github.io/profiles/</a>
41  */
42 @PostConstruct
43 public void initApplication() {
44     log.info("Running with Spring profile(s) : {}", Arrays.toString(env.getActiveProfiles()));
45     Collection<String> activeProfiles = Arrays.asList(env.getActiveProfiles());
46     if (activeProfiles.contains(Constants.SPRING_PROFILE_DEVELOPMENT) && activeProfiles.contains(Constants.SPRING_PROFILE_PRO
47         log.error("You have misconfigured your application! It should not run " +
48             "with both the 'dev' and 'prod' profiles at the same time.");
49     }
50     if (activeProfiles.contains(Constants.SPRING_PROFILE_DEVELOPMENT) && activeProfiles.contains(Constants.SPRING_PROFILE_CLO
51         log.error("You have misconfigured your application! It should not" +
52             "run with both the 'dev' and 'cloud' profiles at the same time.");
53   }
54
55 /**
56  * Main method, used to run the application.
57  *
58  * @param args the command line arguments
59  * @throws UnknownHostException if the local host name could not be resolved into an address
60  */
61 public static void main(String[] args) throws UnknownHostException {
62     SpringApplication app = new SpringApplication(JhipsterApp.class);
63     DefaultProfileUtil.addDefaultProfile(app);
64     Environment env = app.run(args).getEnvironment();
65
66 }
```





A feature model documents  
the **features** of a product line  
and their **relationships**.



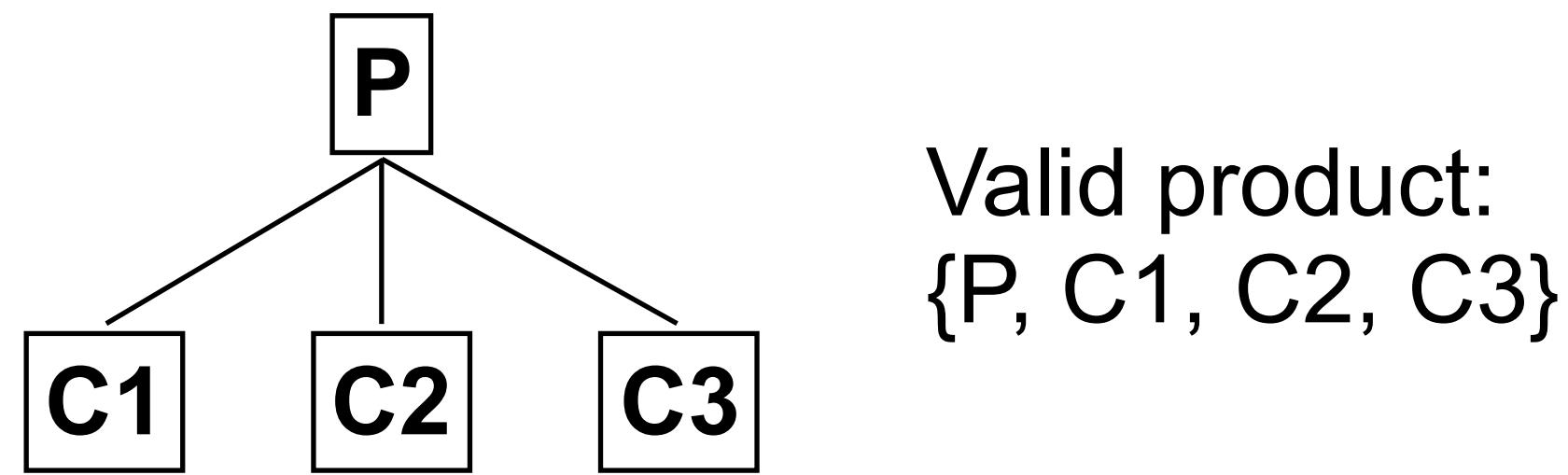
# Vocabulary

Feature = Option

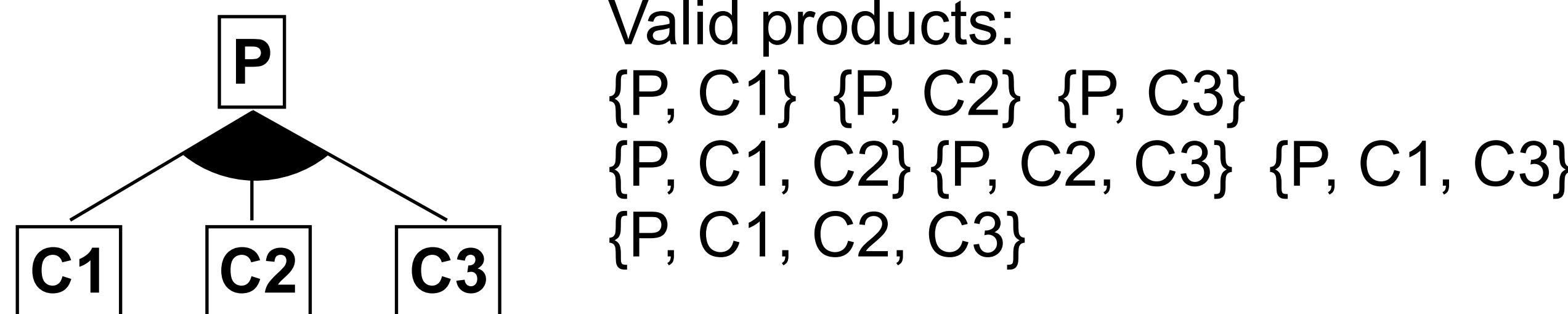
Product = Configuration = Variant

# Feature diagram

- **Tree-like** structure (directed acyclic graph).
- A feature is **decomposed** in sub-features
  - **And** decomposition: if the parent is selected then all sub-features must be selected too

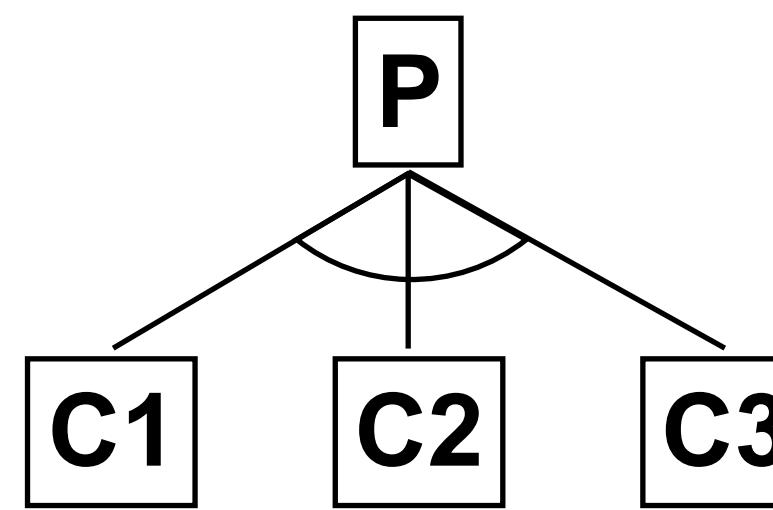


- **Or** decomposition: if the parent is selected then at least one sub-features must be selected too



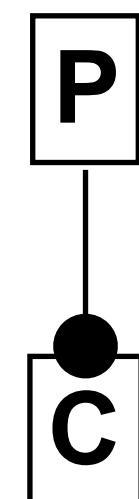
# Feature diagram

- **Tree-like** structure (directed acyclic graph).
- A feature is **decomposed** in sub- features
  - **Xor** decomposition: if the parent is selected then exactly one sub-features must be selected too



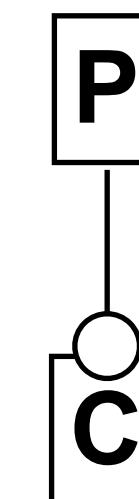
Valid products:  
{P, C1}  
{P, C2}  
{P, C3}

- **Mandatory** feature: if the parent is selected then the feature must be selected



Valid product:  
{P, C}

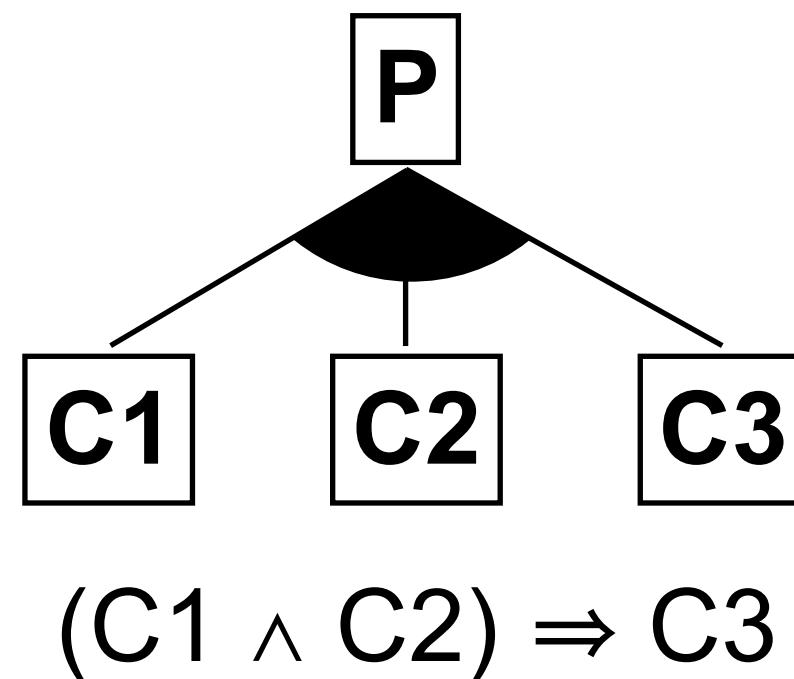
- **Optional** feature: if the parent is selected then the feature must be selected



Valid products:  
{P}  
{P, C}

# Feature diagram

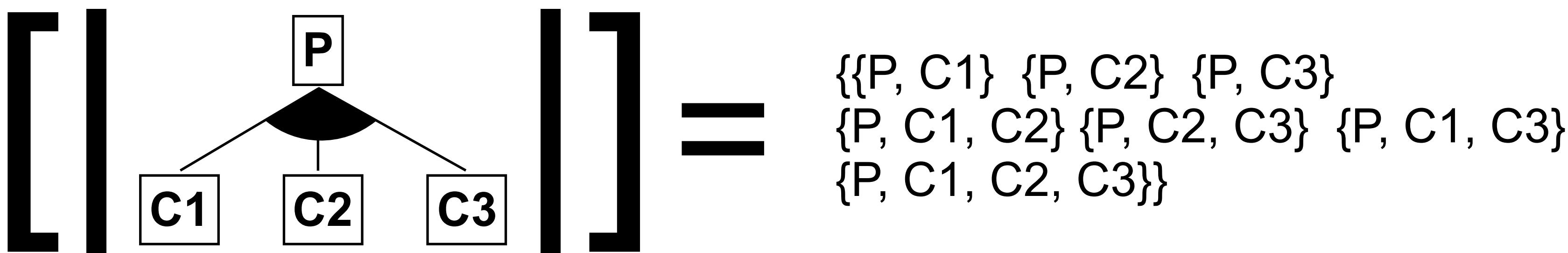
- Cross-tree **constraints** are expressed as boolean expressions over the features.

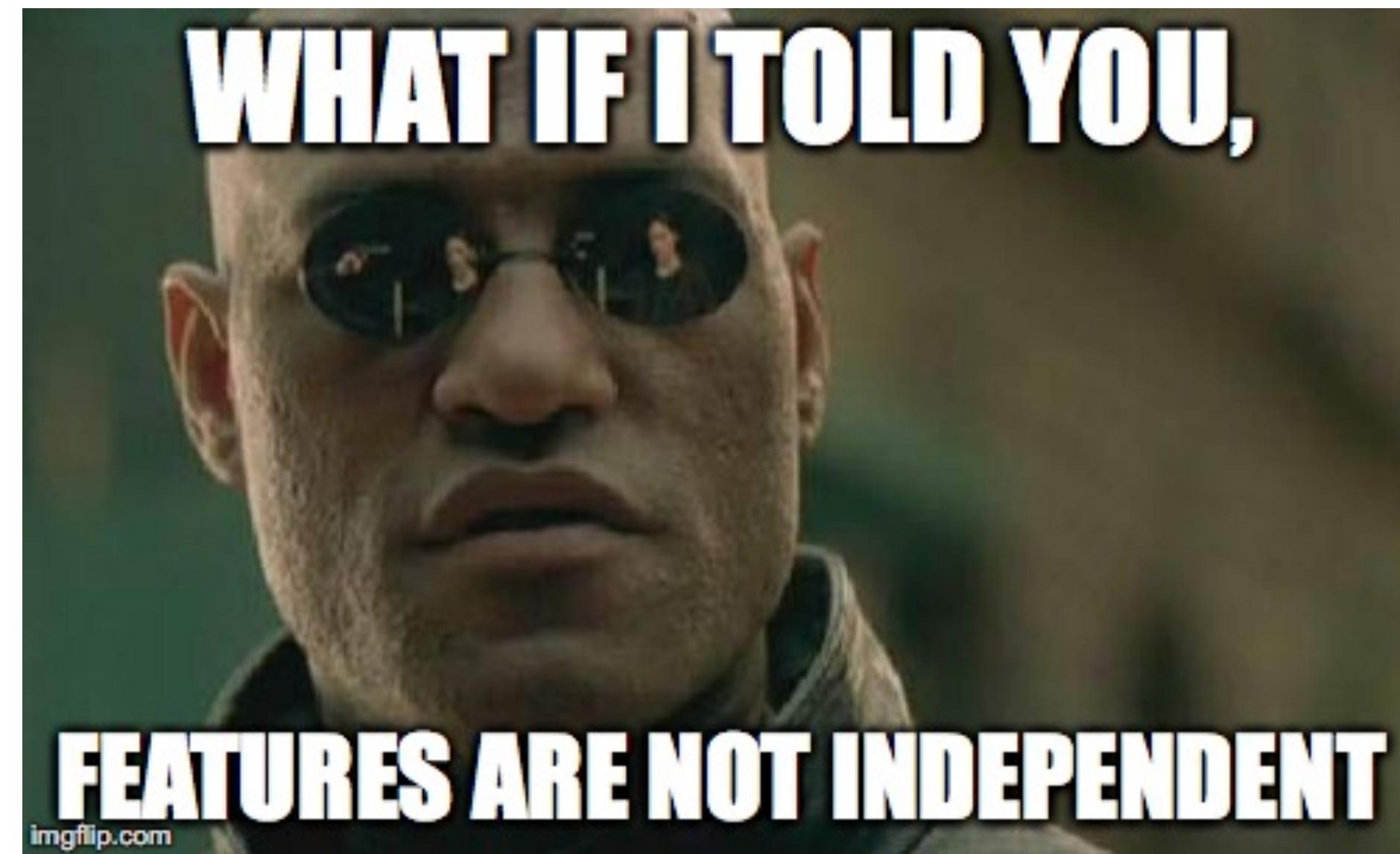


Valid products:

$\{P, C1\}$   $\{P, C2\}$   $\{P, C3\}$   
 $\{\cancel{P, C1, C2}\}$   $\{P, C2, C3\}$   $\{P, C1, C3\}$   
 $\{P, C1, C2, C3\}$

- The **semantics** of a feature model corresponds to all the **valid products** allowed by the feature model.





**Constraints and where to find them**

# Source: Configurator

Application type

Client side options

Server side options

Which type of application would you like to create?

- Monolithic application (recommended for simple projects)
- Microservice application
- Microservice gateway
- JHipster UAA server (for microservice OAuth2 authentication)

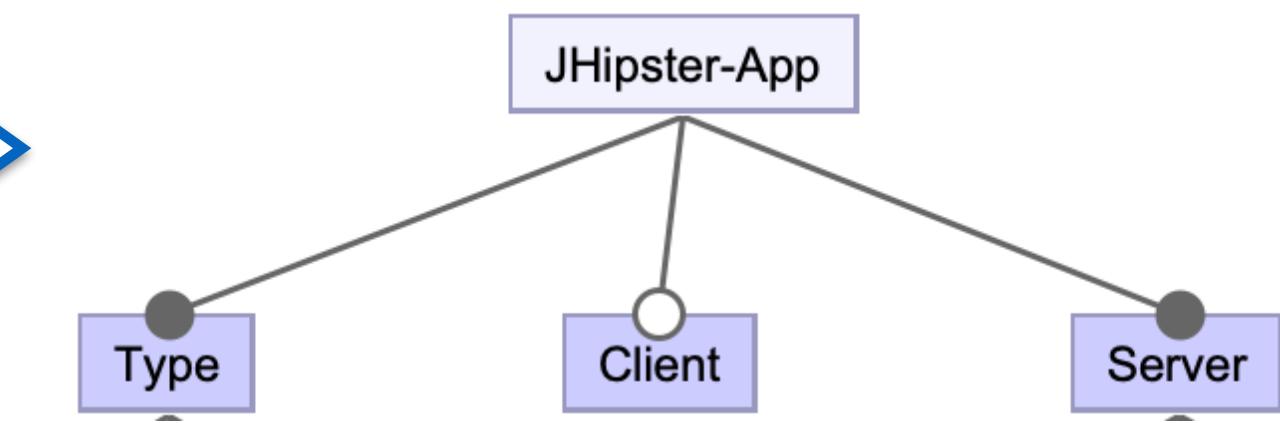
Besides JUnit and Jest, which testing frameworks would you like to use?

- Gatling
- Cucumber
- Protractor

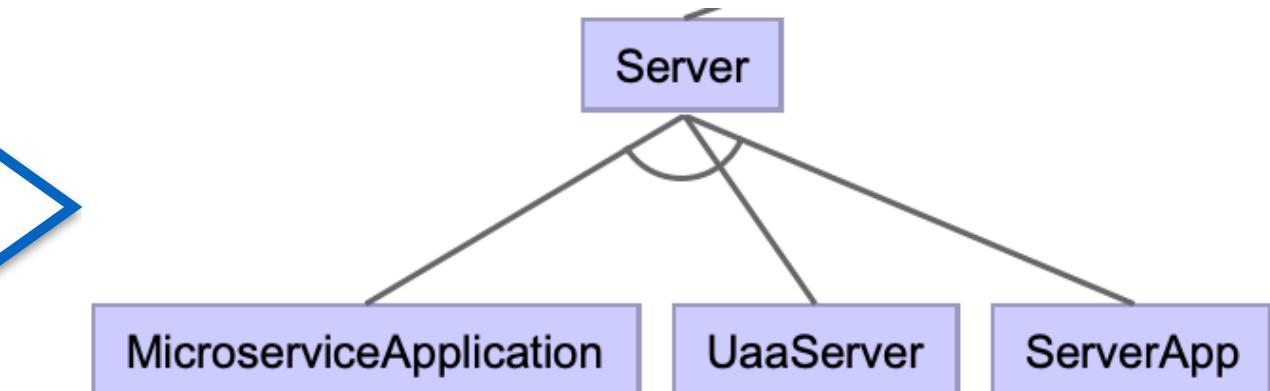
Would you like to enable internationalization support?

- Yes
- No

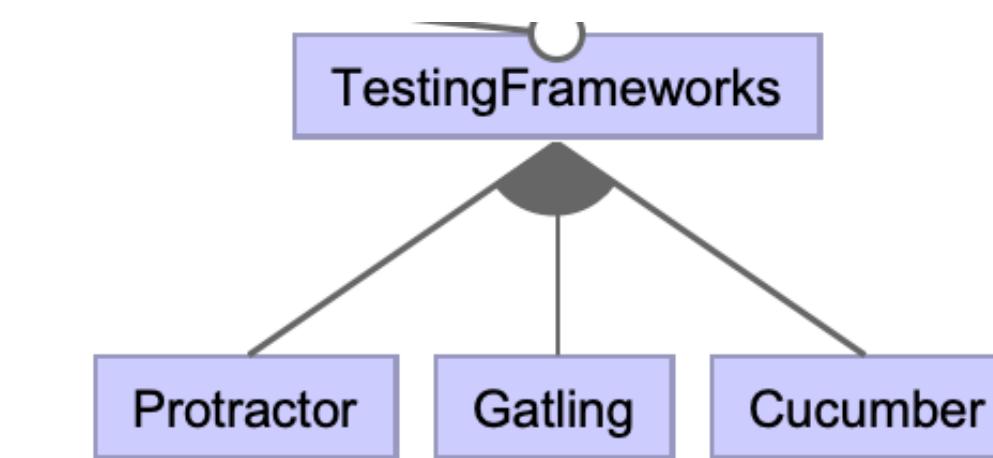
**AND  
decomposition**



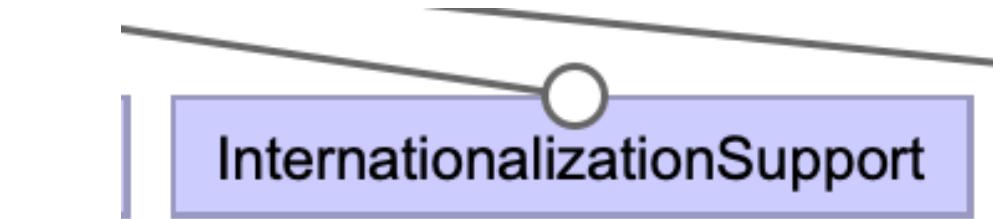
**XOR  
decomposition**



**OR  
decomposition**



**Optional feature**



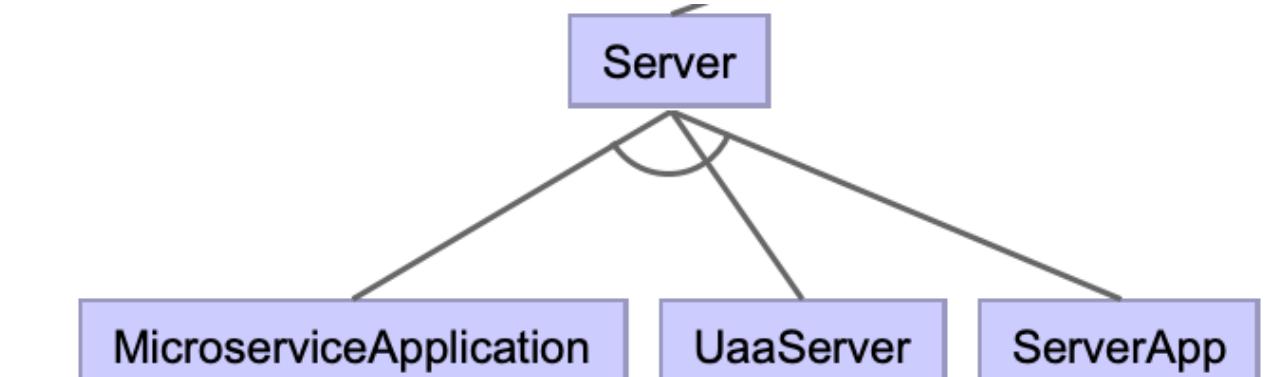
# Source: Documentation

**Which type of application would you like to create?**

You can either use:

- Monolithic application [...]
- Microservice application [...]
- Microservice gateway [...]

XOR decomposition



**Which type of database would you like to use?**

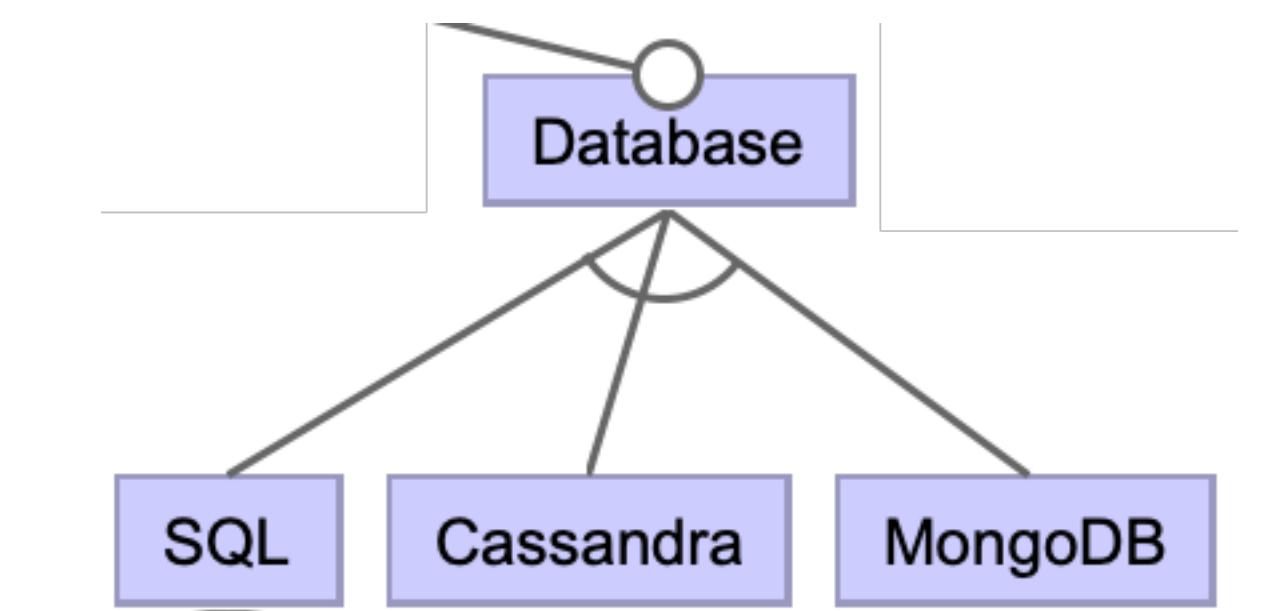
You can choose between:

- No database (only available when using a microservice application)
- An SQL database (H2, MySQL, MariaDB, PostgreSQL, MySQL, Oracle), which you will access with Spring Data JPA
- MongoDB
- Cassandra

Optional feature

XOR decomposition

Additional constraints



$\neg Database \Rightarrow MicroserviceApplication$

$SQL \Rightarrow SpringDataJPA$

**Social login (Google, Facebook, Twitter)**

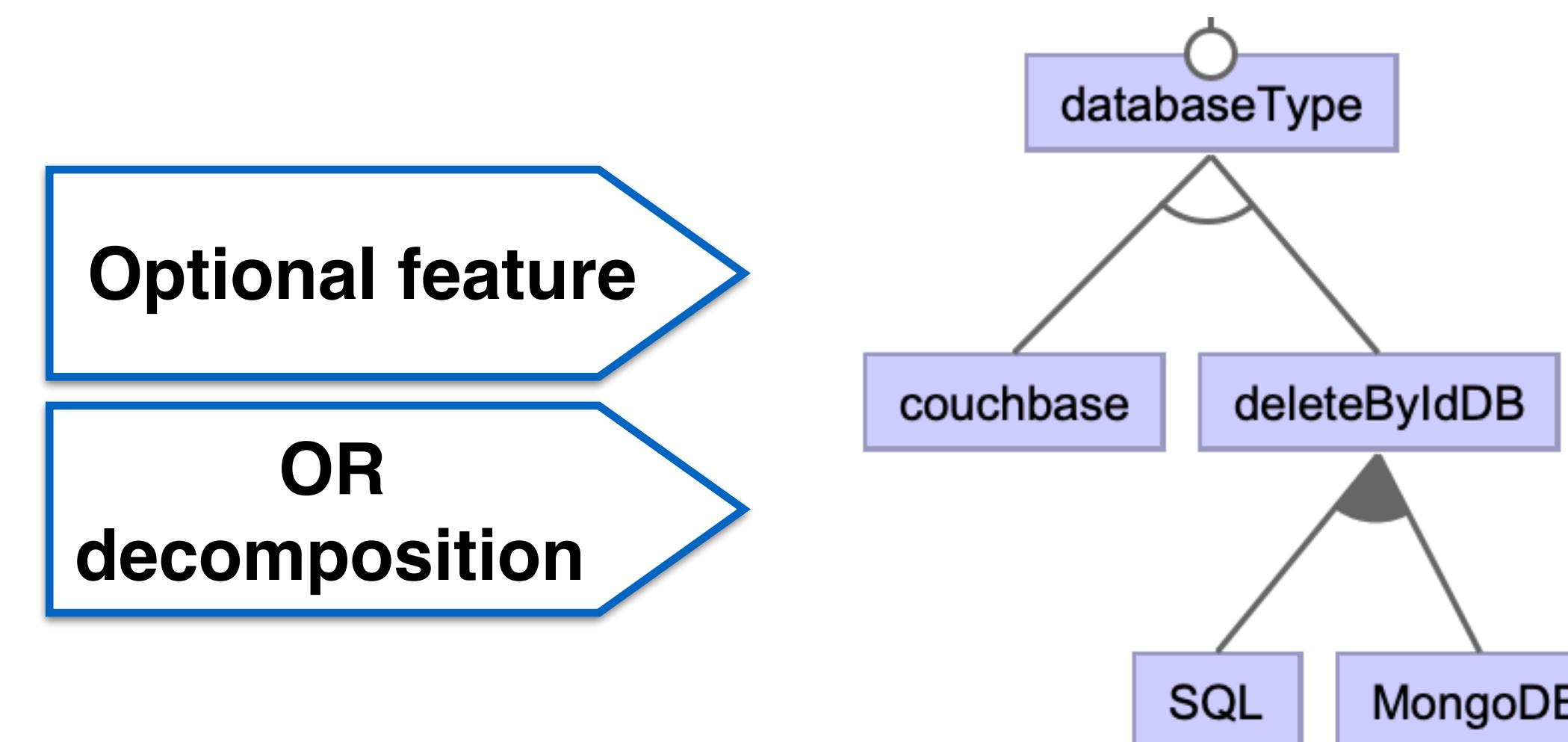
This option is only available if you selected an SQL or MongoDB database.

Additional constraints

$SocialLogin \Rightarrow (SQL \vee MongoDB)$

# Source: Source code

```
public void invalidateSession(@PathVariable String series) throws UnsupportedEncodingException {  
    String decodedSeries = URLDecoder.decode(series, "UTF-8");  
    SecurityUtils.getCurrentUserLogin()  
        .flatMap(userRepository::findOneByLogin)  
        .ifPresent(u ->  
            persistentTokenRepository.findByUser(u).stream()  
                .filter(persistentToken -> StringUtils.equals(persistentToken.getSeries(),  
decodedSeries))  
                <%_ if (databaseType === 'sql' || databaseType === 'mongodb') { _%>  
                    .findAny().ifPresent(t -> persistentTokenRepository.deleteById(decodedSeries)));  
                <%_ } else if (databaseType === 'couchbase') { _%>  
                    .findAny().ifPresent(t -> persistentTokenRepository.deleteBySeries(decodedSeries));  
                <%_ } else { _%>  
                    .findAny().ifPresent(persistentTokenRepository::delete));  
                <%_ } _%>  
} <% } %>
```



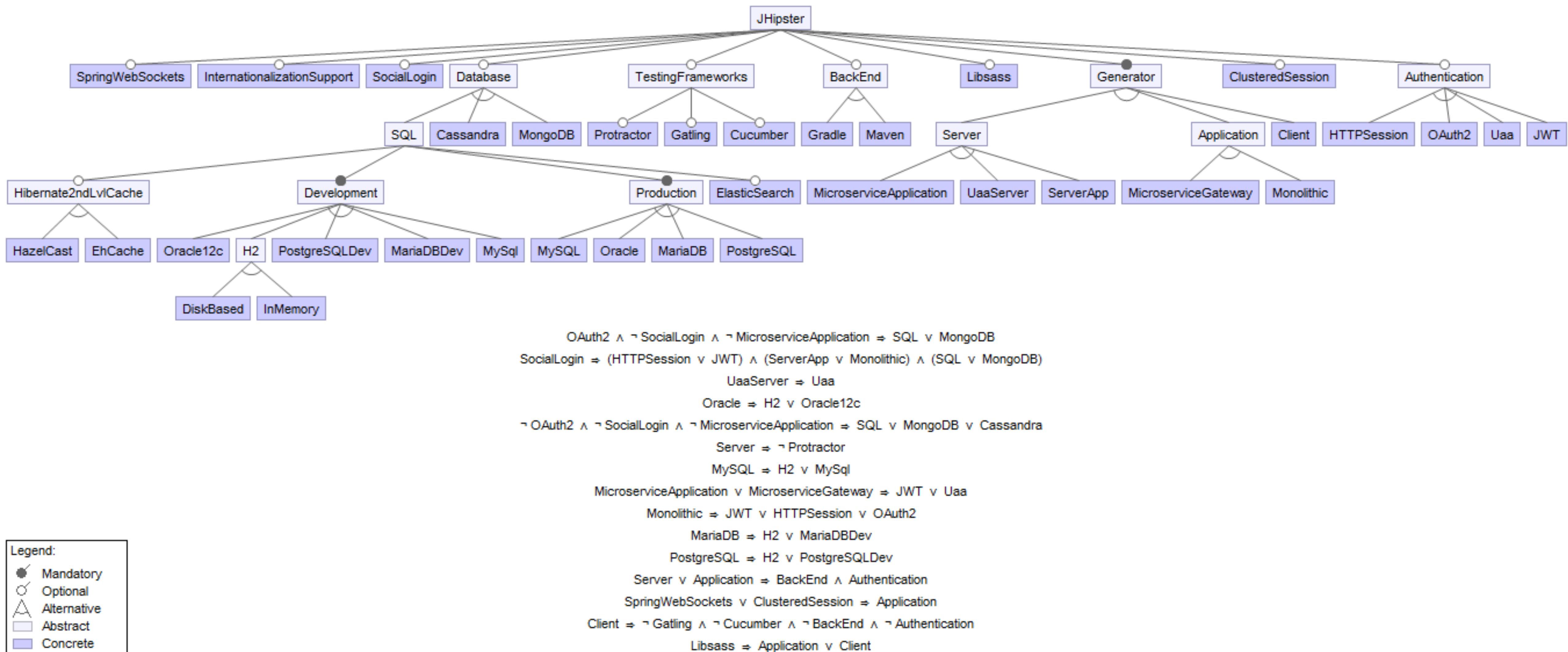
Optional feature

OR  
decomposition

f IDE  
eature

<http://www.featureide.com>

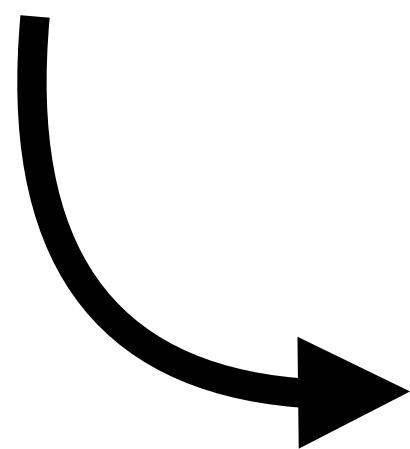
# JHipster v3.6.1



<https://github.com/xdevroey/jhipster-dataset/tree/master/v3.6.1/featuremodel>

# Conjunctive Normal Form (CNF)

JHipster-App  $\wedge$  (Building  $\vee \neg$ Maven)  $\wedge$  (Building  $\vee \neg$ Gradle)  $\wedge$  (Maven  $\vee$  Gradle  $\vee \neg$ Building)  $\wedge$  ( $\neg$ Maven  $\vee \neg$ Gradle)  $\wedge$  (Server  $\vee \neg$ Database)  $\wedge$  (Server  $\vee \neg$ HibernateCache)  $\wedge$  (Server  $\vee \neg$ Building)  $\wedge$  (Building  $\vee \neg$ Server)  $\wedge$  (JHipster-App  $\vee \neg$ Type)  $\wedge$  (JHipster-App  $\vee \neg$ Client)  $\wedge$  (JHipster-App  $\vee \neg$ Server)  $\wedge$  (Type  $\vee \neg$ JHipster-App)  $\wedge$  (Server  $\vee \neg$ JHipster-App)  $\wedge$  (Type  $\vee \neg$ Monolithic)  $\wedge$  (Type  $\vee \neg$ Microservice)  $\wedge$  (Type  $\vee \neg$ Gateway)  $\wedge$  (Monolithic  $\vee$  Microservice  $\vee$  Gateway  $\vee \neg$ Type)  $\wedge$  ( $\neg$ Monolithic  $\vee \neg$ Microservice)  $\wedge$  ( $\neg$ Monolithic  $\vee \neg$ Gateway)  $\wedge$  ( $\neg$ Microservice  $\vee \neg$ Gateway)  $\wedge$  (Database  $\vee \neg$ MongoDB)  $\wedge$  (Database  $\vee \neg$ SQL)  $\wedge$  (Database  $\vee \neg$ Cassandra)  $\wedge$  (MongoDB  $\vee$  SQL  $\vee$  Cassandra  $\vee \neg$ Database)  $\wedge$  ( $\neg$ MongoDB  $\vee \neg$ SQL)  $\wedge$  ( $\neg$ MongoDB  $\vee \neg$ Cassandra)  $\wedge$  ( $\neg$ SQL  $\vee \neg$ Cassandra)  $\wedge$  ( $\neg$ HibernateCache  $\vee$  SQL)  $\wedge$  ( $\neg$ Client  $\vee$  Monolithic)  $\wedge$  (Client  $\vee$  Microservice  $\vee$  Gateway)  $\wedge$  True  $\wedge \neg$ False



- SAT
- Consistency
- Valid configurations
- Dead features
- ...

**JHipster v3.6.1: >26k  
possible products**

# Feature Modelling

- **Compact representation of all products**
  - For instance, using a feature diagram
- **Sources for reverse-engineering**
  - Configurator
  - Documentation
  - Source code
- **Can be formalised for analysis**
  - SAT of a configuration/product
  - Analysis (consistency, dead features, etc.)
  - ...

# Does your application have variability?

Yes, most probably

- **What kind of variability?**
  - Hardware
  - Platform
  - Software
    - Bundles
    - Plugins
    - Command line options
    - Configuration files
    - Microservices
    - ...
- **How is it managed (if it is managed)?**
  - For users
    - Documentation only?
    - Configurator (e.g., JHipster)?
  - For developers
    - Documentation only?
    - Model (e.g., feature model)?
    - Planned in the development lifecycle?



imgflip.com

JAKE-CLARK.TUMBLR

# Variability Binding Time

Variability offers choices.

When we derive a product, we make **decisions**; we decide which features will be included in the product or not. We also say that we **bind** a decision.

# Design-time binding

- Clone and own



# Compile-time Binding

- Decided before or at compile time
- Allows **optimisations**
  - Removes unnecessary code from the product
- Once the product has been generated and deployed, it is **not variable any more**

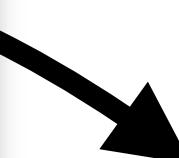
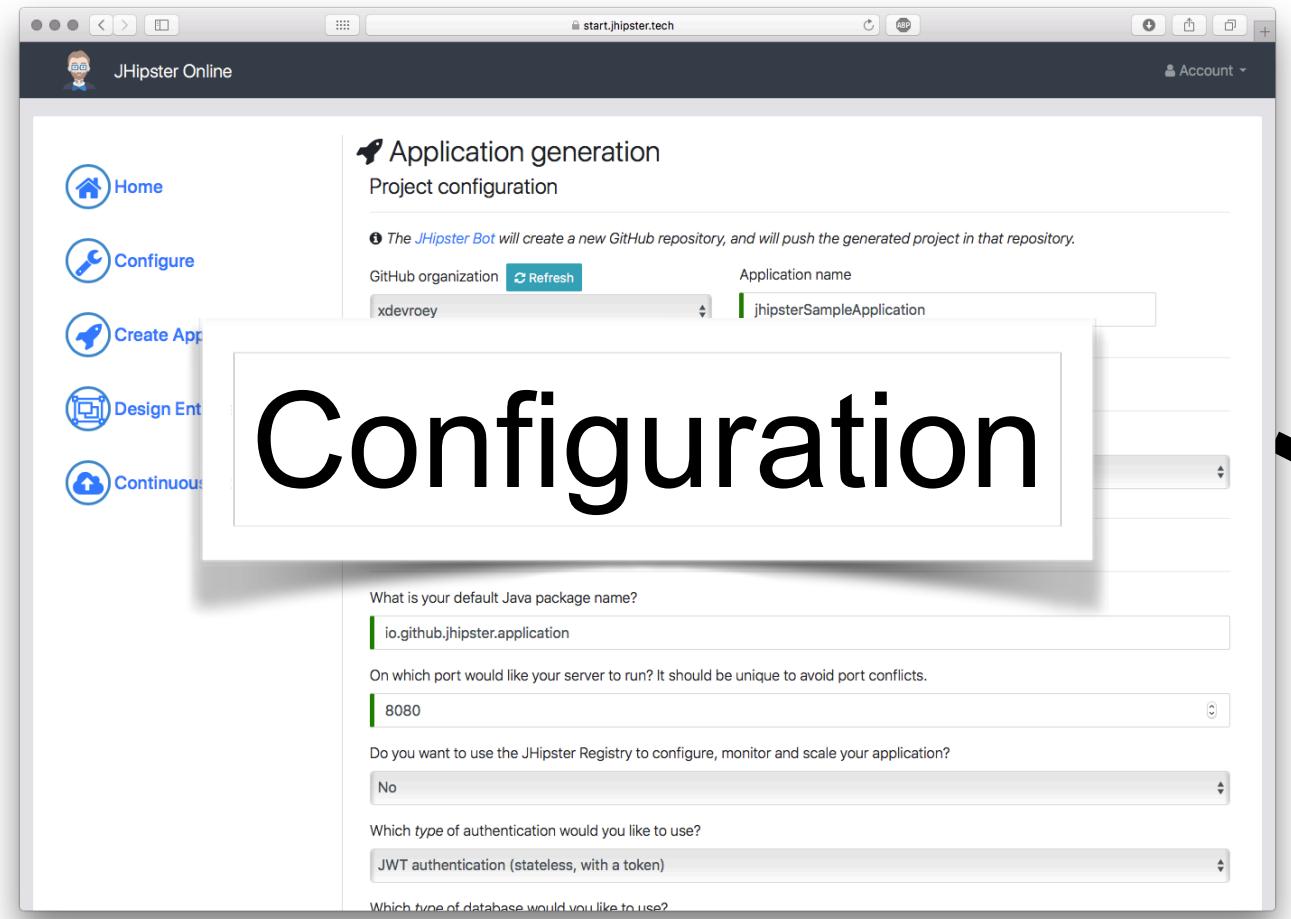
# Conditional compiling

- Implements variability with  **preprocessors**
- Using **annotated** code (here, templates)

```
<%_ if (databaseType === 'mongodb' || databaseType === 'couchbase') { %>←  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-<%=databaseType%></artifactId>←  
</dependency>  
<%_ } %>  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-logging</artifactId>  
</dependency>  
  
<% if (databaseType === 'sql') { %>/**  
 * Spring Data JPA repository for the Authority entity.  
 */<% } %><% if (databaseType === 'mongodb') { %>/**  
 * Spring Data MongoDB repository for the Authority entity.  
 */<% } %><% if (databaseType === 'couchbase') { %>/**  
 * Spring Data Couchbase repository for the Authority entity.  
 */<% } %>  
public interface AuthorityRepository extends <% if (databaseType === 'sql') {  
%>JpaRepository<% } %><% if (databaseType === 'mongodb') { %>MongoRepository<% } %><%  
if (databaseType === 'couchbase') { %>N1qlCouchbaseRepository<% } %><Authority, String>  
{  
}
```



# Conditional compiling

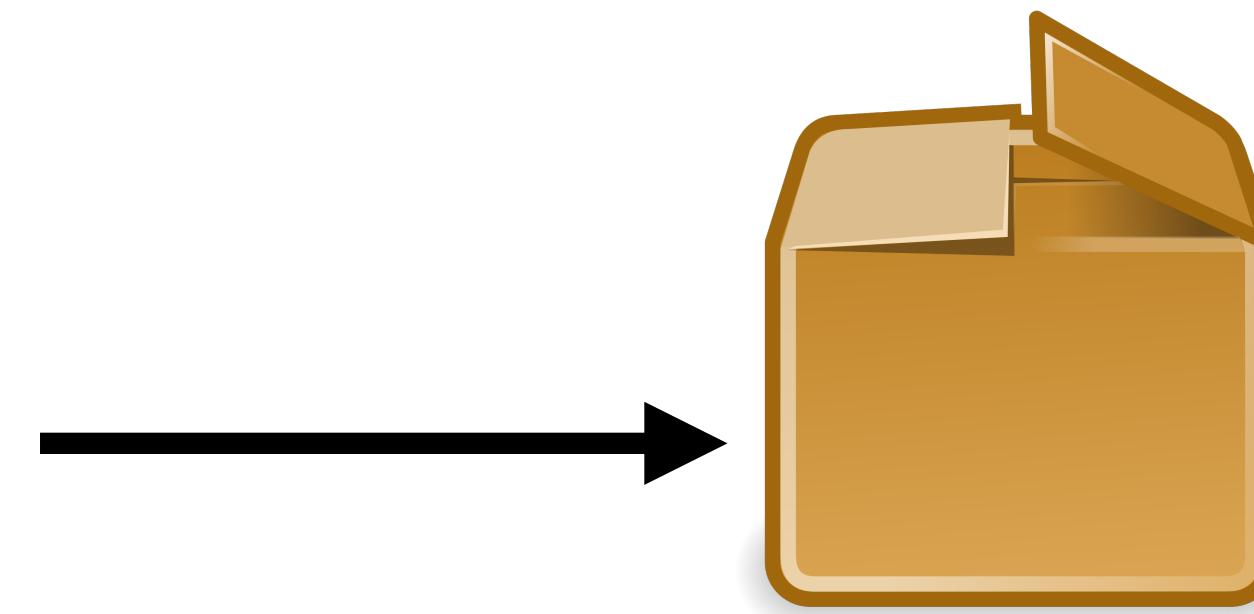
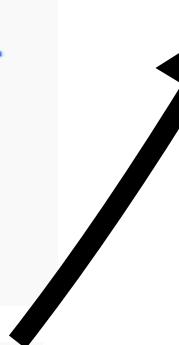


Yeoman

```
<%_ if (databaseType === 'mongodb' || databaseType === 'couchbase') { %>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-<%=databaseType%></artifactId>
</dependency>
<% } %>
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-mongodb</artifactId>
</dependency>
<% if (databaseType === 'couchbase') { %>/*
 * Spring Data Couchbase repository for the Authority entity.
 */<% } %>
<% if (databaseType === 'couchbase') { %>/*
 * Spring Data Couchbase repository for the Authority entity.
 */<% } %>
public interface AuthorityRepository extends <% if (databaseType === 'sql') {
  >JpaRepository<% } %><% if (databaseType === 'mongodb') { %>MongoRepository<% } %><%
if (databaseType === 'couchbase') { %>N1qlCouchbaseRepository<% } %><Authority, String>
{
}
```

A screenshot of annotated code showing conditional compilation logic using JHipster annotations. The code includes sections for MongoDB and Couchbase databases, with specific repository implementations for each.

Annotated code



Source code  
without annotations



# Load-time Binding

- Decided after compilation when the program is started
- More flexibility for **reconfiguration**
  - ... but Requires a reboot
- Memory and performance **overhead**
  - All variations are compiled into a single binary
  - Consistency conditions must be checked at load time
  - Unnecessary functionality may be considered as potential **security threat**

# Parameters and Configuration Files

- Implement variability using conditional statements (such as if and switch)
- Controlled by configuration parameters
  - Passed to the method or module,
  - Set as global variables in a system

Gcc

```
/* Enable -Werror=coverage-mismatch when -Werror and -Wno-error
   have not been set. */
if (!global_options_set.x_warnings_are_errors
    && warn_coverage_mismatch
    && (global_dc->classify_diagnostic[OPT_Wcoverage_mismatch] ==
         DK_UNSPECIFIED))
  diagnostic_classify_diagnostic (global_dc, OPT_Wcoverage_mismatch,
                                 DK_ERROR, UNKNOWN_LOCATION);

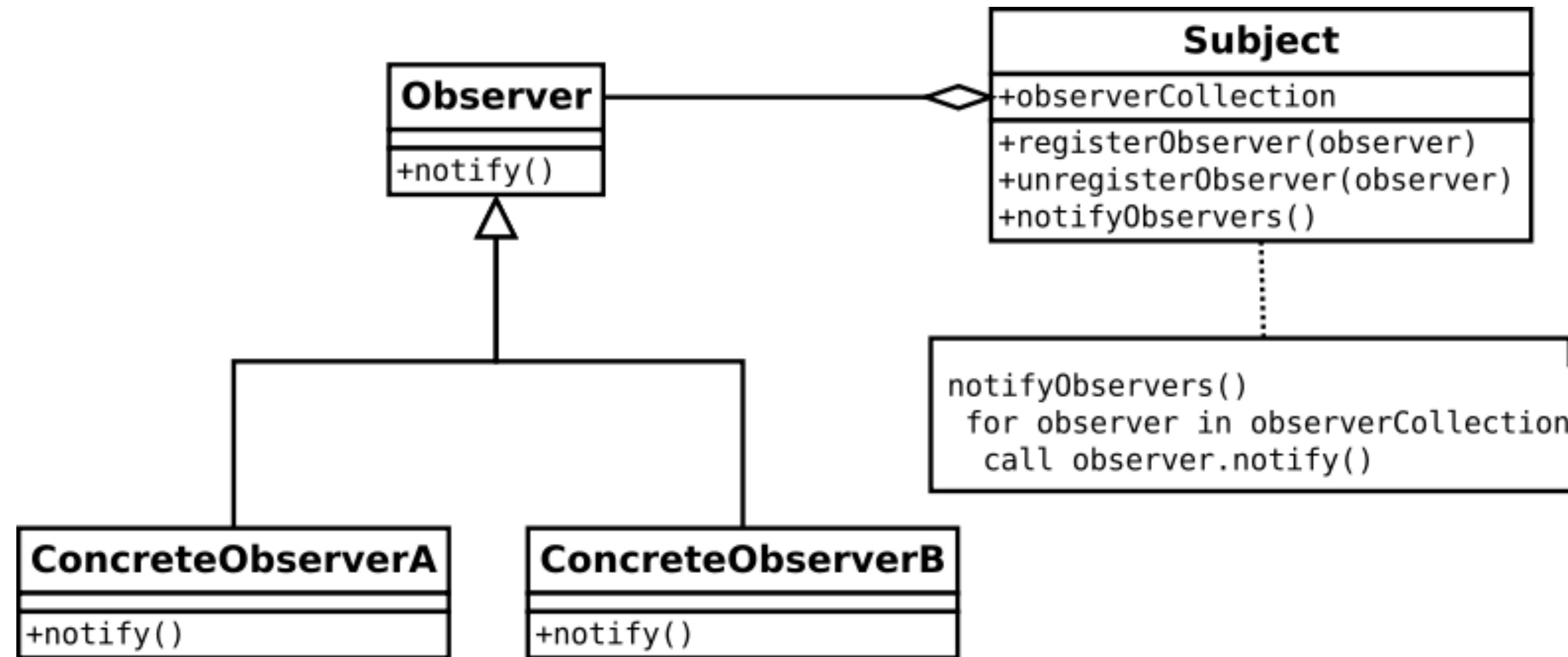
/* Save the current optimization options. */
optimization_default_node = build_optimization_node (&global_options);
optimization_current_node = optimization_default_node;
```

# Run-time Binding

- Decided during program execution and may be changed latter
  - Used in dynamic adaptive systems (e.g., robots)
- Most flexible to **reconfigure**
  - Able to react to internal and external stimuli by adapting its behaviour
- Memory and performance **overhead**
  - All variations are compiled into a single binary
  - Consistency conditions must be checked at run time
- Unnecessary functionality may be considered as potential **security threat**

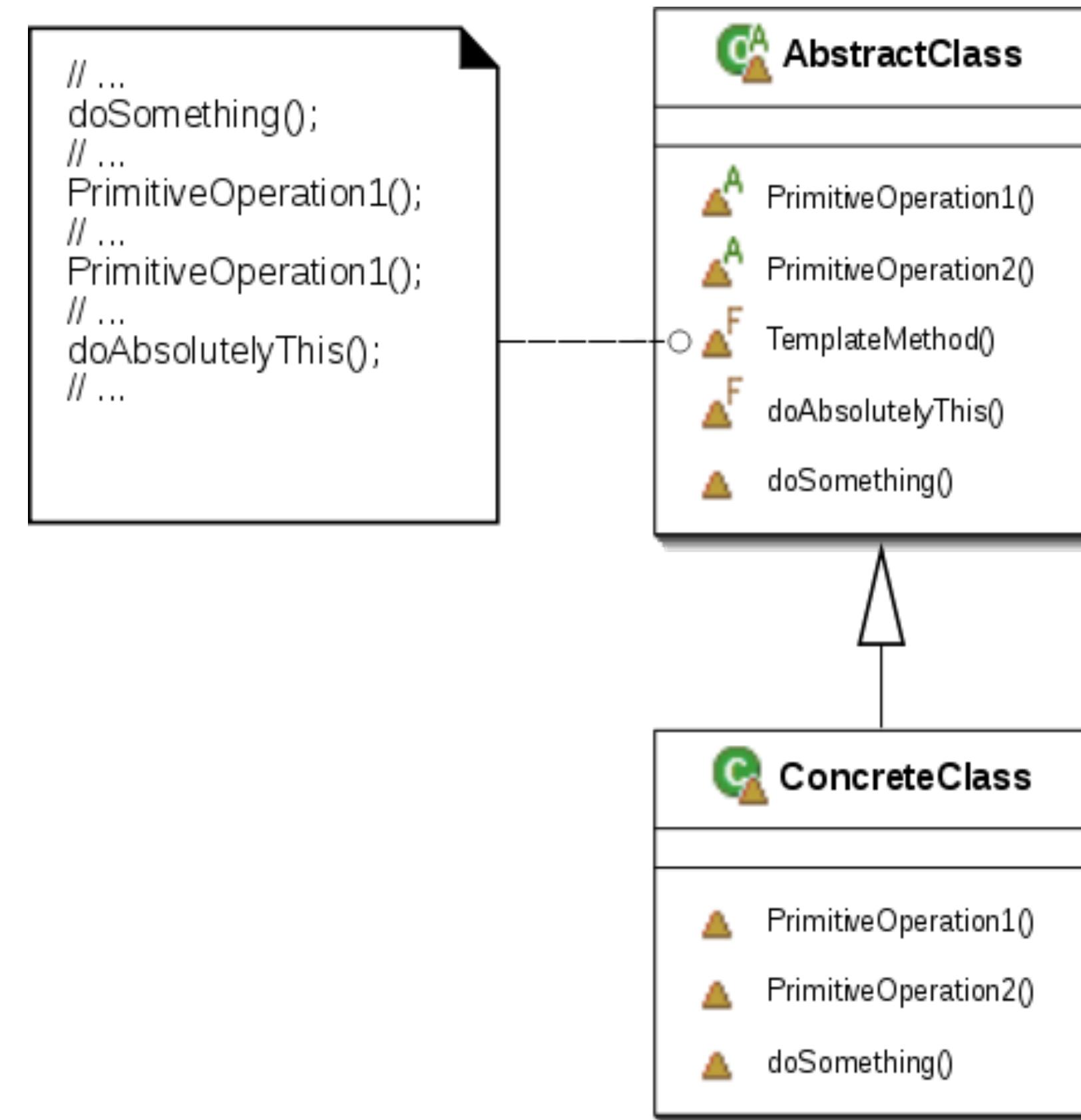
# Design Patterns

- Observer pattern (publish/subscribe) to implement distributed event handling
  - Feature are implemented as observers
  - Variability is achieved by registering or not registering observers



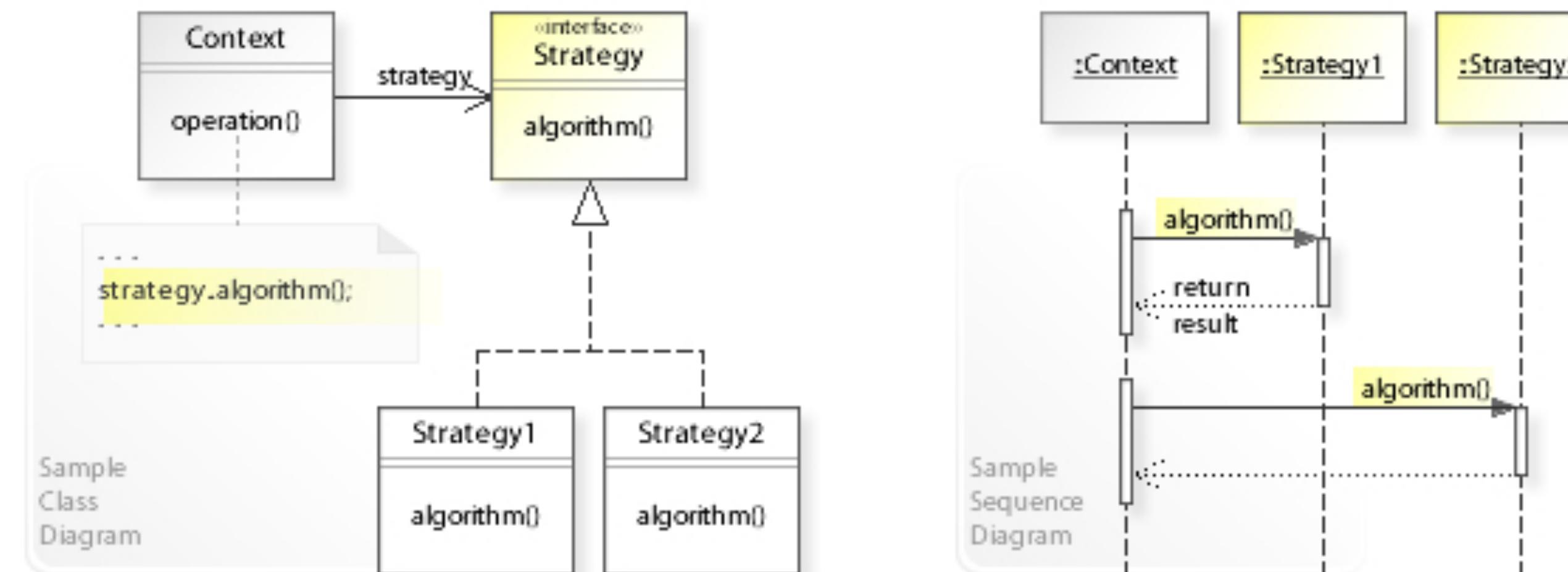
# Design Patterns

- Template-method pattern
  - Using an abstract class
  - Implement feature's behaviour using inheritance



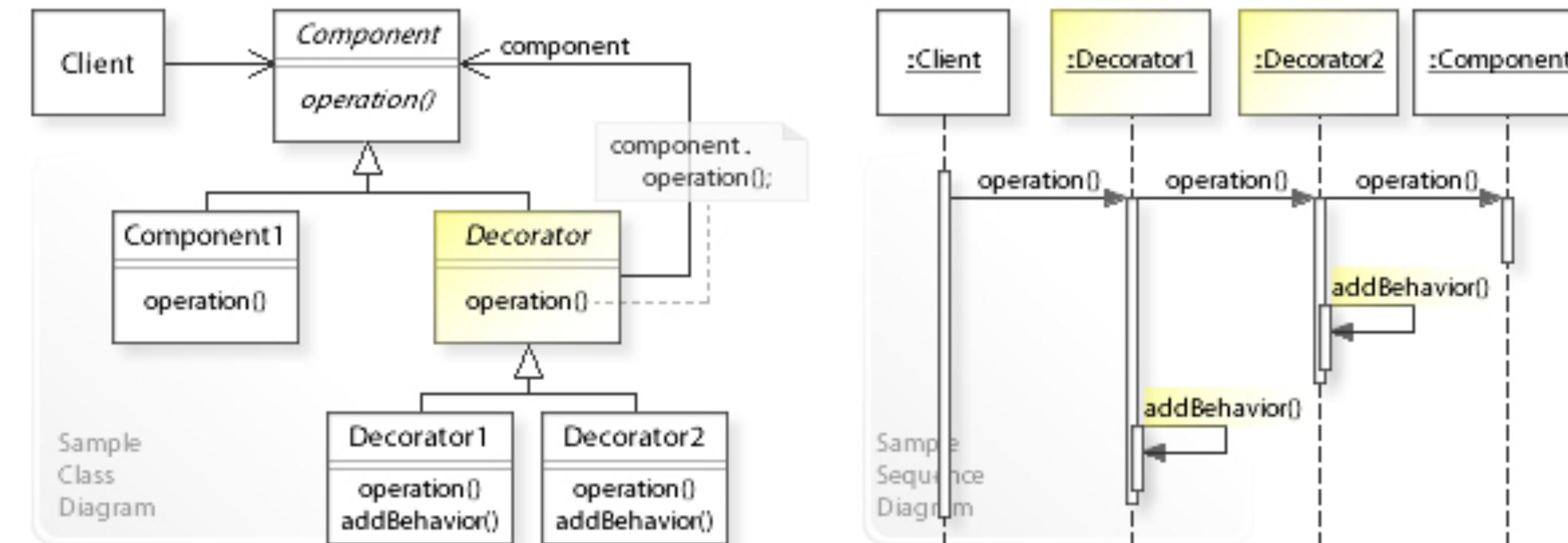
# Design Patterns

- Strategy pattern
  - Encodes a callback mechanism
  - Alternative features where each corresponds to one implementation of the algorithm
  - Uses polymorphism instead of conditional statements



# Design Patterns

- Decorator pattern
  - Extends object with additional behaviour at run time
  - Optional features and feature groups, of which multiple features can be selected, are implemented as decorators



# Variability Binding Time

- **Design-time**

- Clone and own

- **Compile-time**

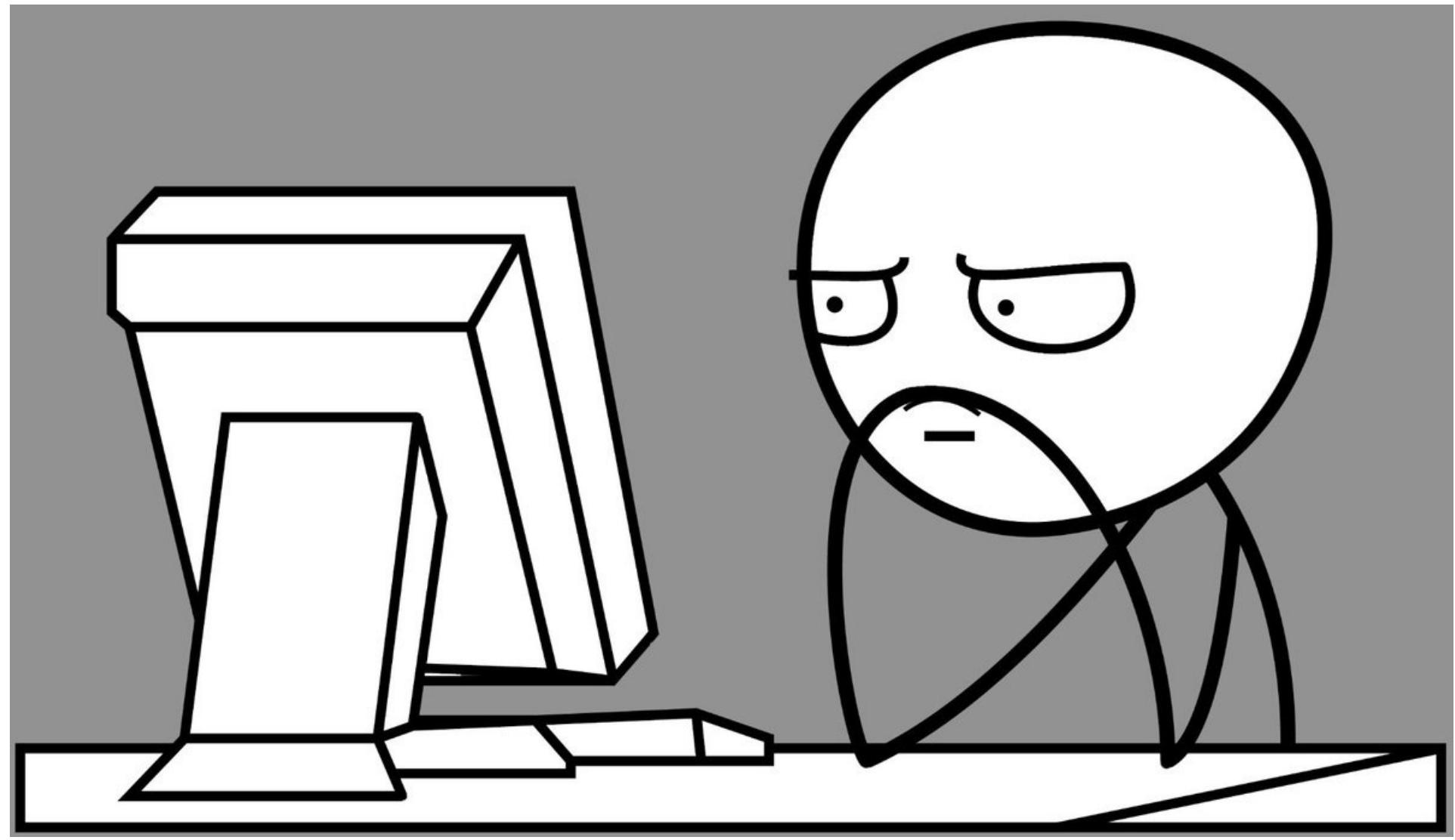
- Conditional compilation

- **Load-time**

- Program parameters and configuration files

- **Run-time**

- Design patterns



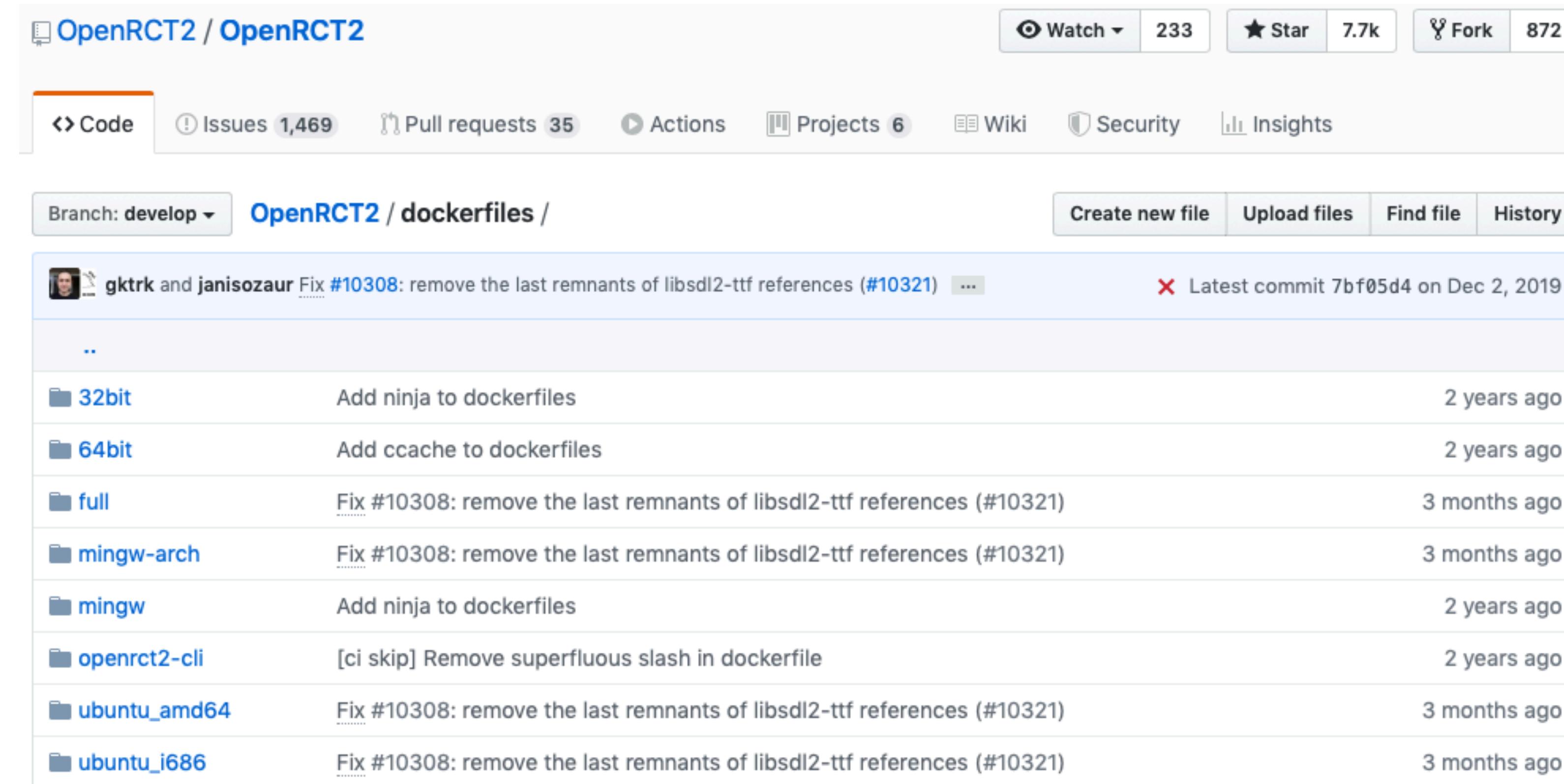
**Take-away**

# Does your application have variability?

Yes, most probably

- **What kind of variability?**
  - Hardware
  - Platform
  - Software
    - Bundles
    - Plugins
    - Command line options
    - Configuration files
    - Microservices
    - ...
- **How is it managed (if it is managed)?**
  - For users
    - Documentation only?
    - Configurator (e.g., JHipster)?
  - For developers
    - Documentation only?
    - Model (e.g., feature model)?
    - Planned in the development lifecycle?

# Example: Docker images



The screenshot shows the GitHub repository page for `OpenRCT2 / OpenRCT2`. The repository has 233 watchers, 7.7k stars, and 872 forks. The `Code` tab is selected, showing 1,469 issues and 35 pull requests. Other tabs include Actions, Projects (6), Wiki, Security, and Insights.

The current branch is `develop`, and the specific commit history for the `OpenRCT2 / dockerfiles` folder is displayed. The latest commit was made by `gktrk and janisozaur` on Dec 2, 2019, fixing #10308: remove the last remnants of libsdl2-ttf references (#10321). The commit message also mentions `#10321`.

The commit history lists the following changes:

- 32bit: Add ninja to dockerfiles (2 years ago)
- 64bit: Add ccache to dockerfiles (2 years ago)
- full: Fix #10308: remove the last remnants of libsdl2-ttf references (#10321) (3 months ago)
- mingw-arch: Fix #10308: remove the last remnants of libsdl2-ttf references (#10321) (3 months ago)
- mingw: Add ninja to dockerfiles (2 years ago)
- openrct2-cli: [ci skip] Remove superfluous slash in dockerfile (2 years ago)
- ubuntu\_amd64: Fix #10308: remove the last remnants of libsdl2-ttf references (#10321) (3 months ago)
- ubuntu\_i686: Fix #10308: remove the last remnants of libsdl2-ttf references (#10321) (3 months ago)

# Variability Management section outline

- Examples of variability in your system, and their manifestation in architectural discussions
  - **Variability management**
  - **(Partial) feature model**
  - Current variability **implementation mechanism** of your system
    - Conditional compiling
    - Conditional statements (if-then-else)
    - Design patterns
  - **Variability binding time**
    - Design-time
    - Compile-time
    - Load-time
    - Run-time

# Take-aways

- Variability enables **adaptability** to the business domain
  - It should be **planned** and **managed**
  - Identify development **practices**
    - Clone and own
    - Software product line
    - Most probably something intermediate
    - Binding
      - Design, compile, load, and/or runtime binding
      - Variability implementation mechanism(s) (conditional compiling, parameters, design patterns)
  - **Feature modelling** helps to manage variability
    - Analysis on the feature model
    - Sample products to test