APACHECON
Europe

16-18.11.16

SEVILLE, SPAIN

How to Manage a Nearly One Million SLOCs Project

Emmanuel Lécharny, Software architect, Symas
Apache Member

THE LINUX FOUNDATION

Disclaimer

No images were harmed in the making of these slides (hopefully)

(Attribution done when needed)

1 Million lines of code :

1 Million lines of code :

"That's going to be

HUUUUUGE!"

How big are projects ?

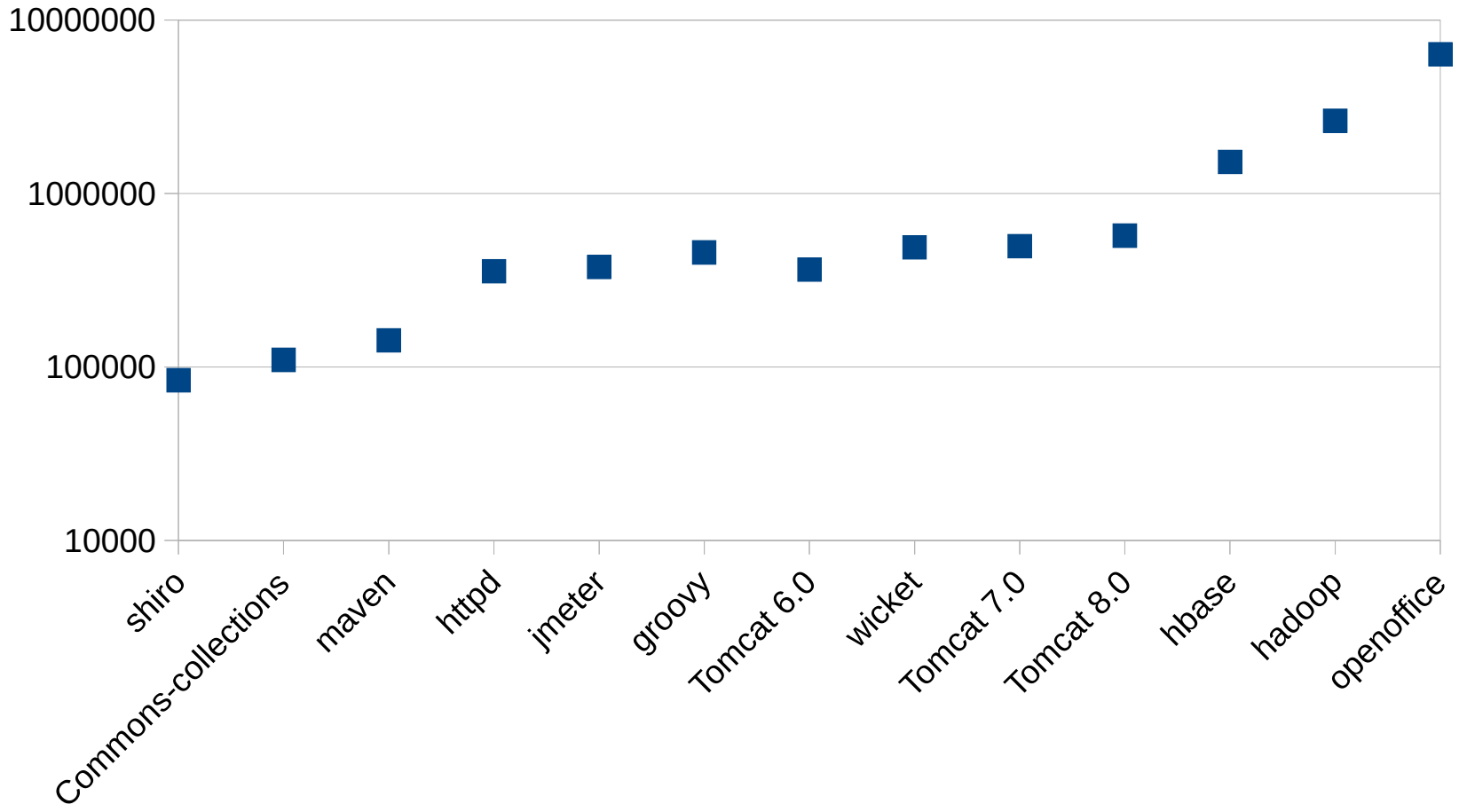
Actually, quite big...

Definition

SLOCs : Single Line Of Code

I.e. not a comment, not a blank line

Shiro :	83 959 slocs
Commons-collections :	109 838 slocs
Maven :	142 257 slocs
HTTPd :	356 333 slocs
Jmeter :	377 054 slocs
Groovy :	457 307 slocs
Tomcat 6.0 :	364 561 slocs
Wicket :	489 115 slocs
Tomcat 7.0 :	496 602 slocs
Tomcat 8.0 :	571 801 slocs
...	...
Hbase :	1 520 477 slocs
Hadoop :	2 626 856 slocs
Openoffice :	6 347 083 slocs


10000000
1000000
100000
10000
shiro
Commons-collections
maven
httpd
jmeter
groovy
Tomcat 6.0
wicket
Tomcat 7.0
Tomcat 8.0
hbase
hadoop
openoffice

Back in 1980... :

"Managing a few thousands lines of assembly language is the limit.

25 000 lines of C is another limit."

Back in the 1980's :

- Barely no IDE
- Very limited debugging tools
- Version control, anyone ?
- Unit/Integration tests, WTF ?
- No standard OS
- No mails (almost)
- BBS to share code

2000 :

~~Y2K bug !~~

Internet all over !

AND

Open Source Software

Get used to it :

1 Million single lines of code is not a mammoth project,

not anymore

So, what are our options ?


ONE WAY

Let me introduce you to

The Directory project

But it could be any

any other project !

Started in 2003

TLP in 2005 (22th Apache project)

54 committers over the last
10 years

Numbers

	Slocs	Comments	Blanks	
Apacheds	227,361	112,533	71,314	
Bulkloader	2,694	N/A	N/A	
Fortress-commander	16,023	4,912	3,421	
Fortress-core	68,888	36,065	11,776	
Fortress-enmasse	4,627	7,539	1,393	
Fortress-realm	872	1,388	356	
JDBM	15,431	N/A	N/A	
Directory Client	4,698	8,185	1,383	
Mavibot	25,073	N/A	N/A	
MINA	47,316	32,093	12,151	
Shared	217,407	112,883	59,459	
Studio	225 490	125,066	60,158	
Total	**855,880**	440,664+	221,411+	**~1,500,000**

And counting...

Blank lines increase the code **readability !**

Comments are mainly Javadoc (which is code !)

They really

help

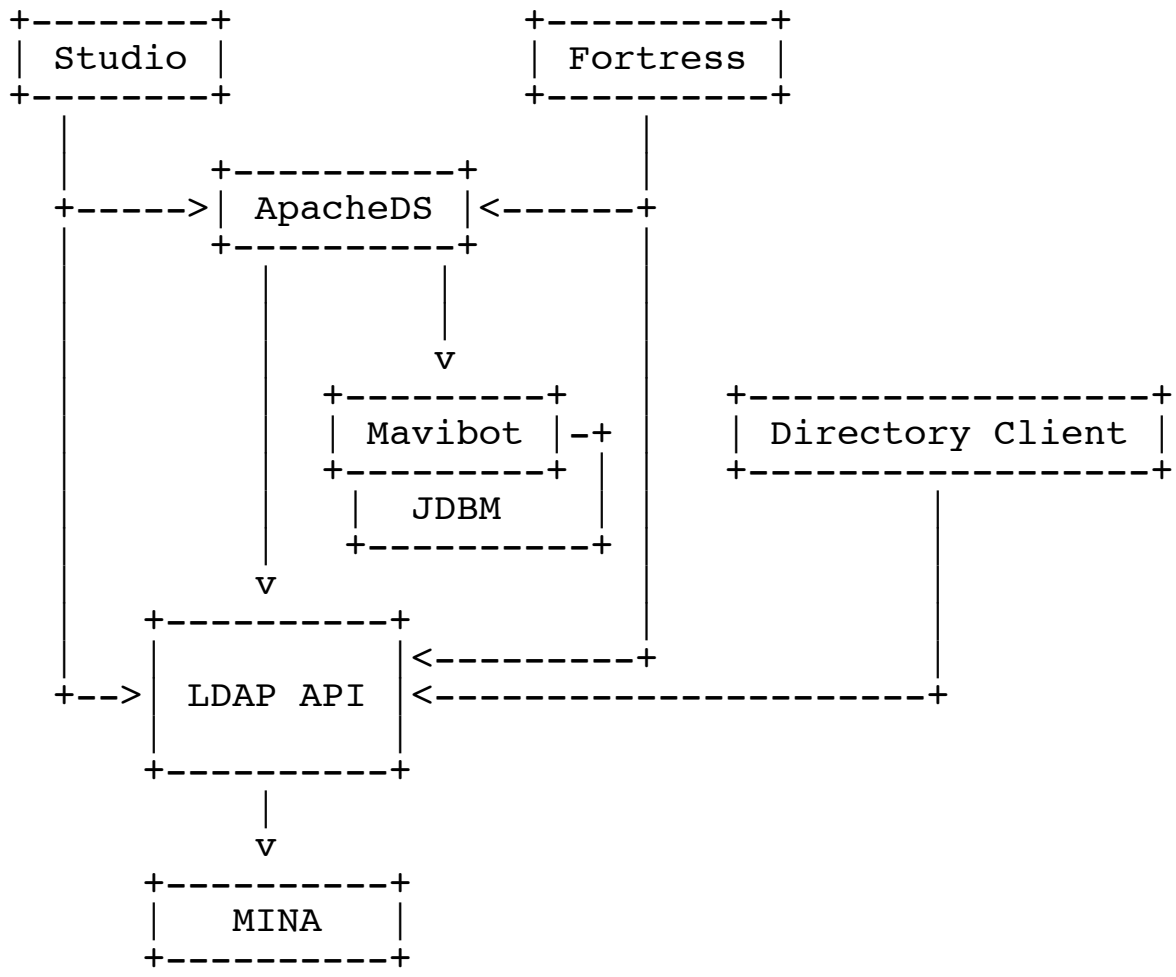
your readers.

Divide to conquer !

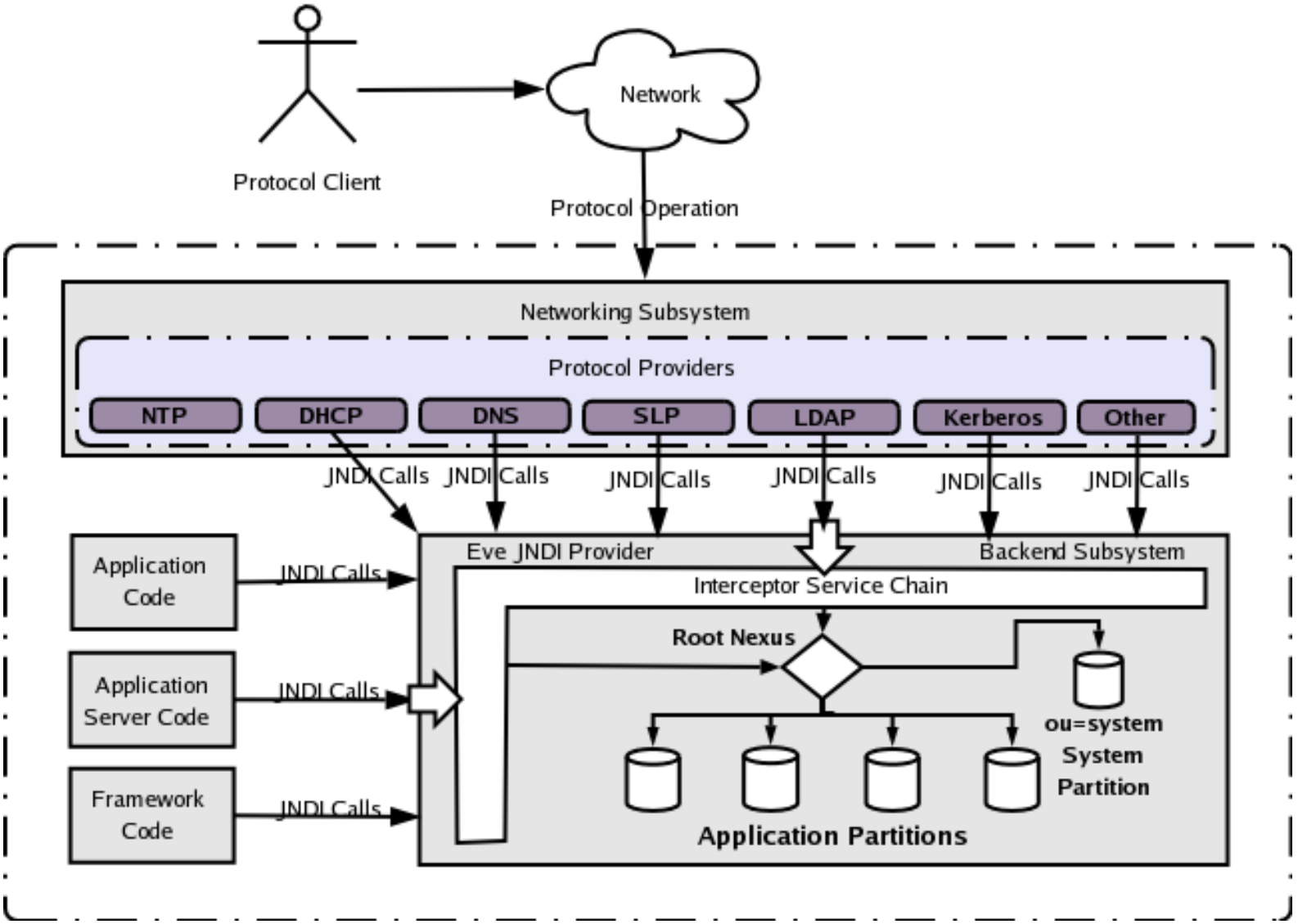
Modularization

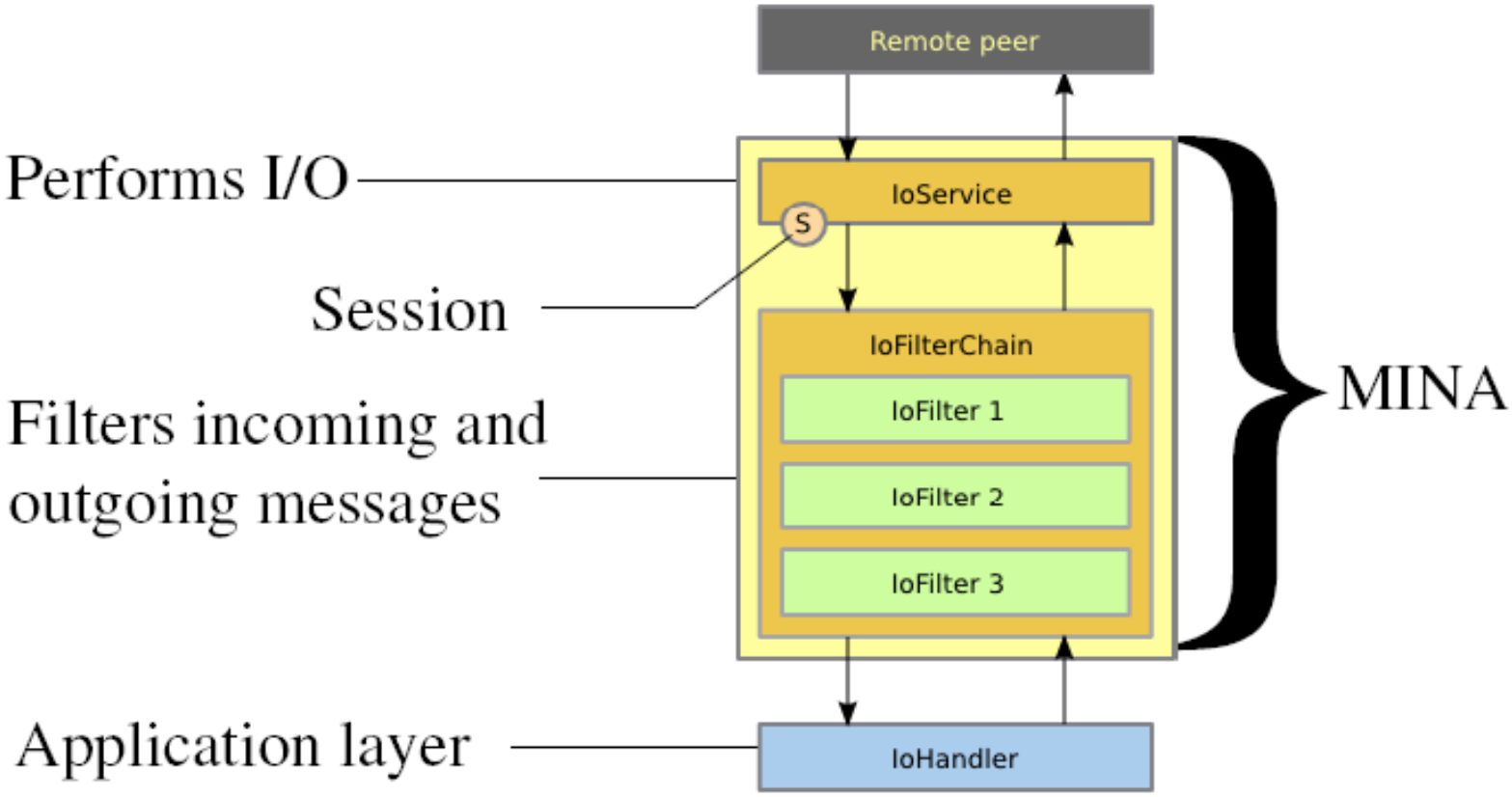
is

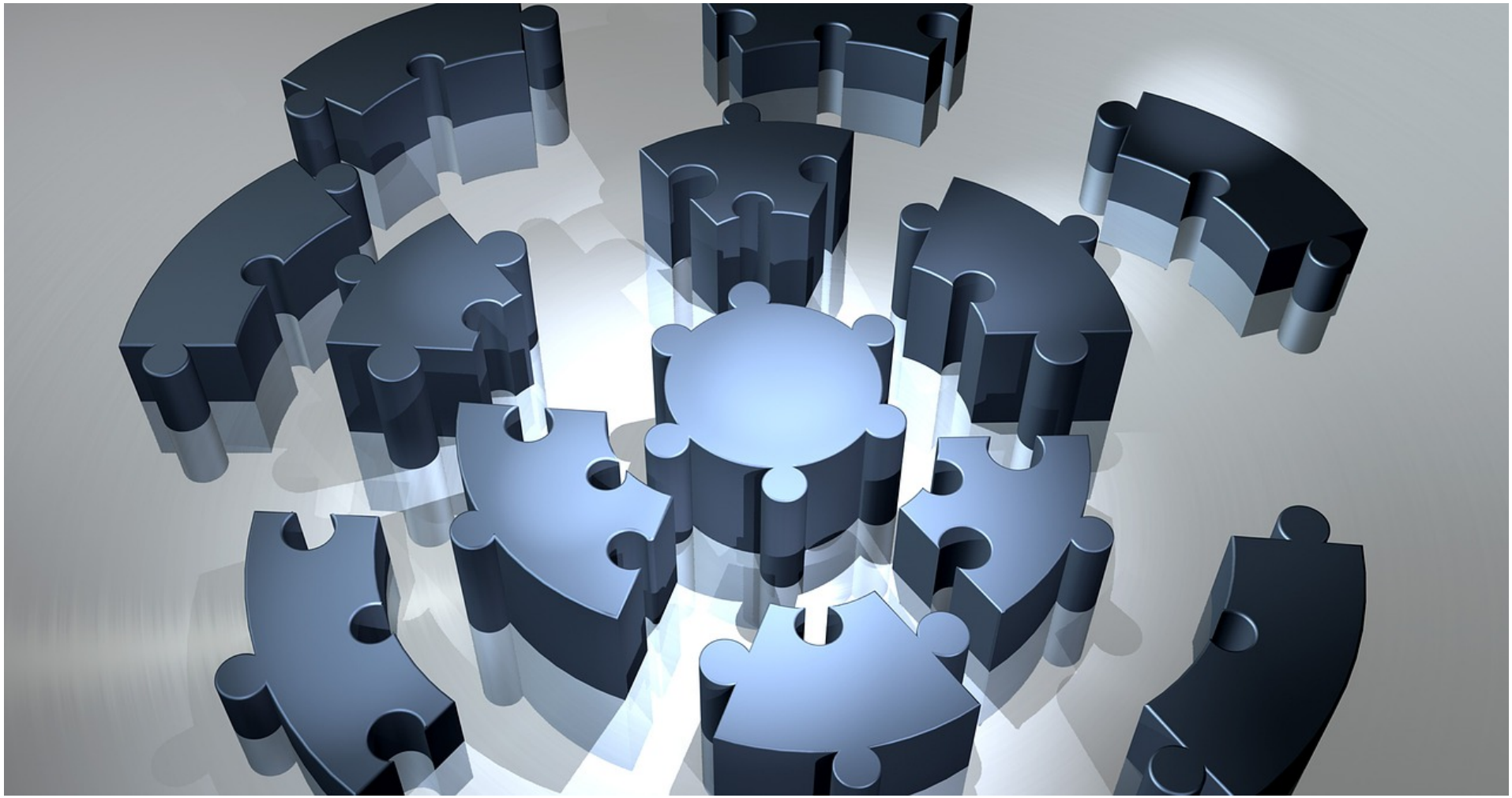
the key


Studio
Fortress
ApacheDS
Mavibot
JDBM
Directory Client
LDAP API
MINA


Layers,
Layers,
Layers !!!


Network
Protocol Client
Protocol Operation
Networking Subsystem
Protocol Providers
NTP
DHCP
DNS
SLP
LDAP
Kerberos
Other
JNDI Calls
JNDI Calls
JNDI Calls
JNDI Calls
JNDI Calls
JNDI Calls
Application Code
Application Server Code
Framework Code
JNDI Calls
JNDI Calls
JNDI Calls
Eve JNDI Provider
Backend Subsystem
Interceptor Service Chain
Root Nexus
ou=system
System
Partition
Application Partitions


Remote peer
Performs I/O
IoService
S
Session
IoFilterChain
IoFilter 1
Filters incoming and
outgoing messages
IoFilter 2
IoFilter 3
MINA
Application layer
IoHandler



Plugins to the rescue !

In ApacheDS :

- Controls
- Extended operations
- Interceptors
- Backends
- Stored procedures
- Authenticators
- Schema elements (comparators, Normalizers)

...

OSGi™ ?

Java 9 ?

Code

Make it

READABLE !!!

(rosetta stone, By © Hans Hillewaert, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=3153928)

* It's going to be written once,
  read many times
* Don't expect anyone to write
  the javadoc : do it !
* Use significant names !
* VT100 is over, so is the 80 char
* Optimize later : it's not urgent

Respect the code : it's likely not yours...

Use a code formatter

don't change the code if not needed

Comments : pay your respect
to readers, and to yourself :

2 years from now on, you'll
be happy to find a comment...

A pre-defined formatter will save you endless discussions...

Format on save, but keep comments off it !

Names : you are not a compiler, your memory is limited...

Make it simpler for readers, ban two-letter name from your code

(supercalifragilisticexpialidocious)

Use some code checkers
(Checkstye, SonarQube, ...) :
they will save your life
more than once !

Added bonus : make your code
consistent...

Use a decent IDE.

Vi is not an option in 2016.

It must handle your build tool.

Talking about build tools :

When it works, don't
Try to fix it !

Have your team endorse it

Use a decent one...

Version control

Today, it's GIT.
Don't fight.
CVS died, too...

IP/License

It's too late when it's release time !

Due diligence is mandatory.
Do it early.

Installers

Mandatory when your project is not an API

WIKI is saving you a lot of time in the long run.

What's the point in explaining things more than once ?

Mailing List

The place where things happen.

Use it, use it, use it...

Experiment

Branches, sub-projects...
Keep people interested !

Rewrite

We don't use Latin anymore !


COMMUNITY

The hardest part... and the most important one !

* Attract new people
* Handle newcomers
* Deal with absentees
* Top guns and average people

People come and go,

deal with it !

EGO

It's fueling your project,
it might burn it to ashes.

Try to be humble
(from time to time...)

Users are your asset
and your best PRs.

Respect them, help them.

Roles :

You can't master the whole code base.

Focus on your part ! Make it easy to jump in...

Contributions

Welcome them, but be careful :

* IP
* Quality
* Hidden agenda
* Maintenance

Anticipate !

You are going to
maintain code
you didn't wrote

Be kind and

respectful.

But know when to say NO.

BDLF

does not fly
in the very long run...

Advertise !

* Conferences
* Blogs
* Announces
* Tweets
* Good looking web site
* Whatever makes your project shine...

Release OFTEN !

Turn the Lights On your project

(Shangaï Circus show, http://marc-francoise.eklablog.com/cirque-a62000627)

Thank you !