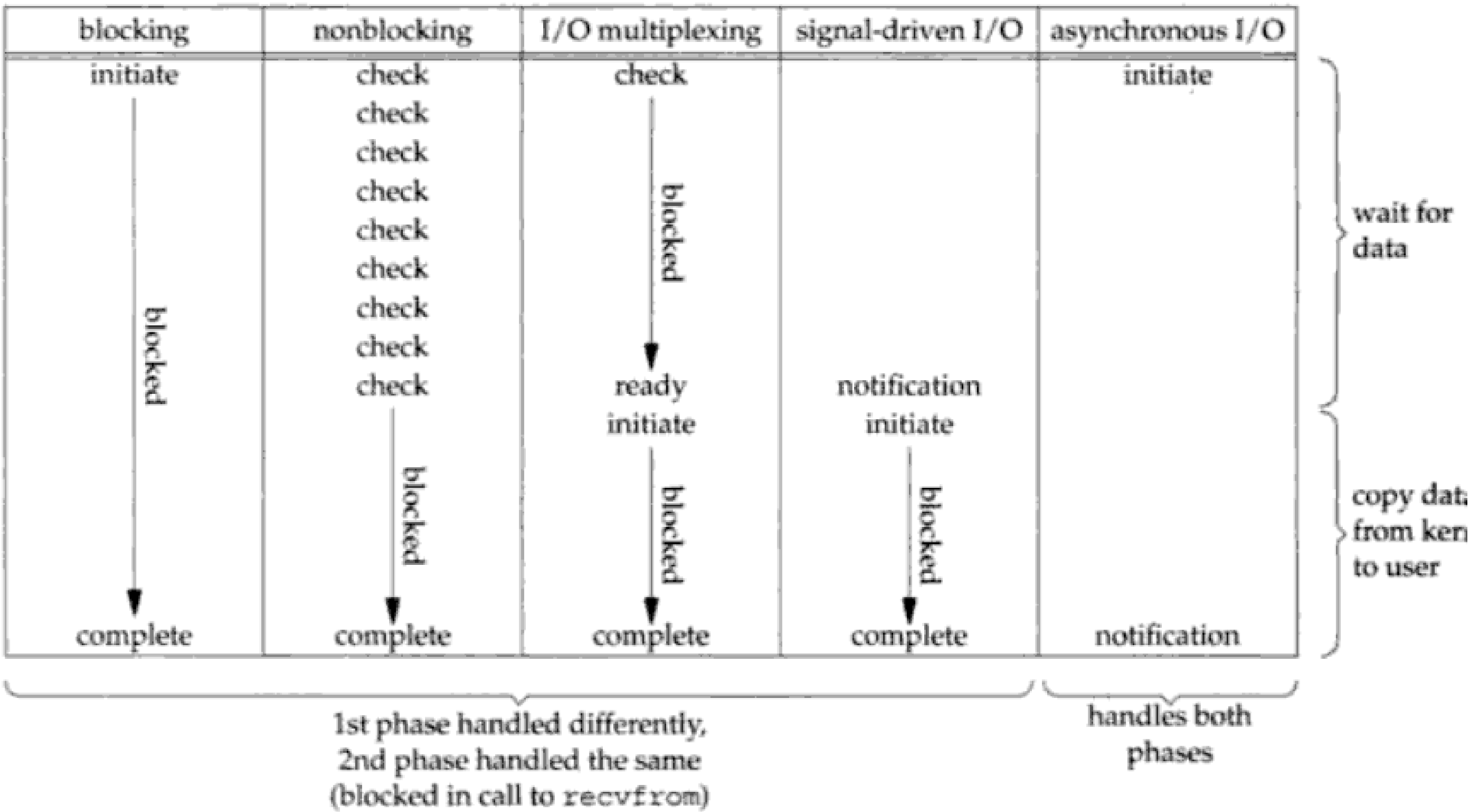
Play2从Netty说起

张胜杰

Agenda

- I/O Model
- Netty
- Reactive web application
- Play2

five I/O models(Richard Stevens,Unix Network Programming)



blocking	nonblocking	I/O multiplexing	signal-driven I/O	asynchronous I/O
initiate	check	check		initiate
	check			
	check			
	check	blocked		
	check			
	check			
blocked	check			
	check			
	check	ready	notification	
		initiate	initiate	
	blocked	blocked	blocked	
complete	complete	complete	complete	notification

wait for data

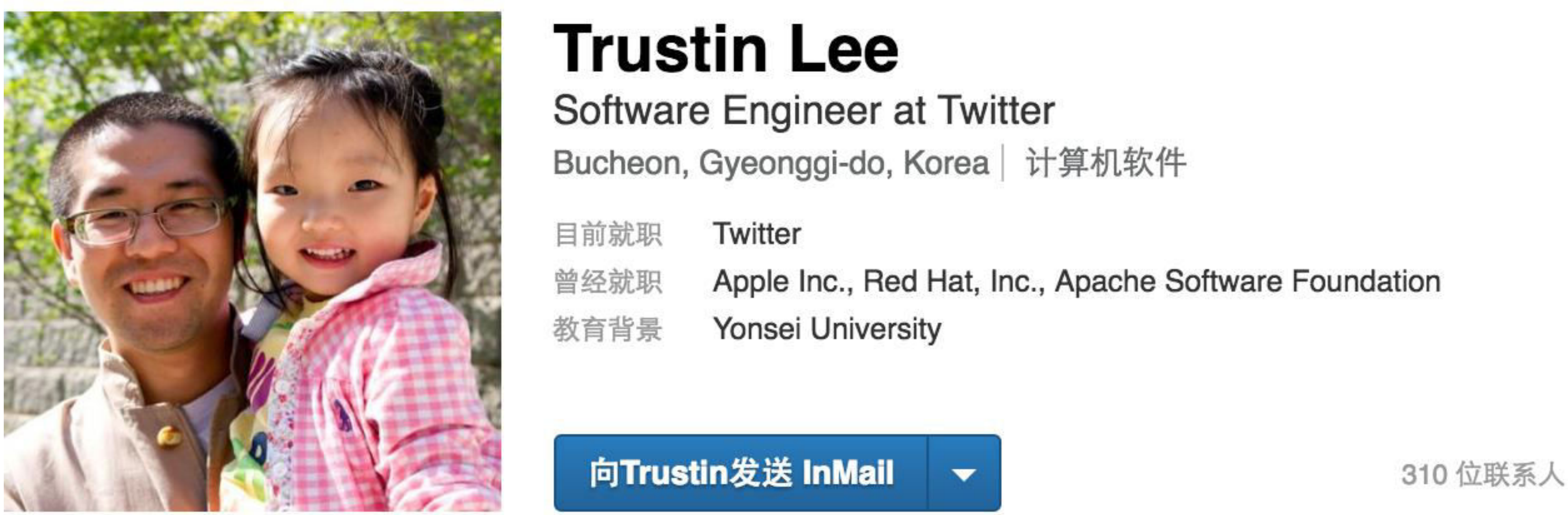
copy dat from ker to user

1st phase handled differently,
2nd phase handled the same
(blocked in call to `recvfrom`)

handles both
phases

What’s Netty

Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients



Trustin Lee

Software Engineer at Twitter

Bucheon, Gyeonggi-do, Korea | 计算机软件

目前就职	Twitter
曾经就职	Apple Inc., Red Hat, Inc., Apache Software Foundation
教育背景	Yonsei University

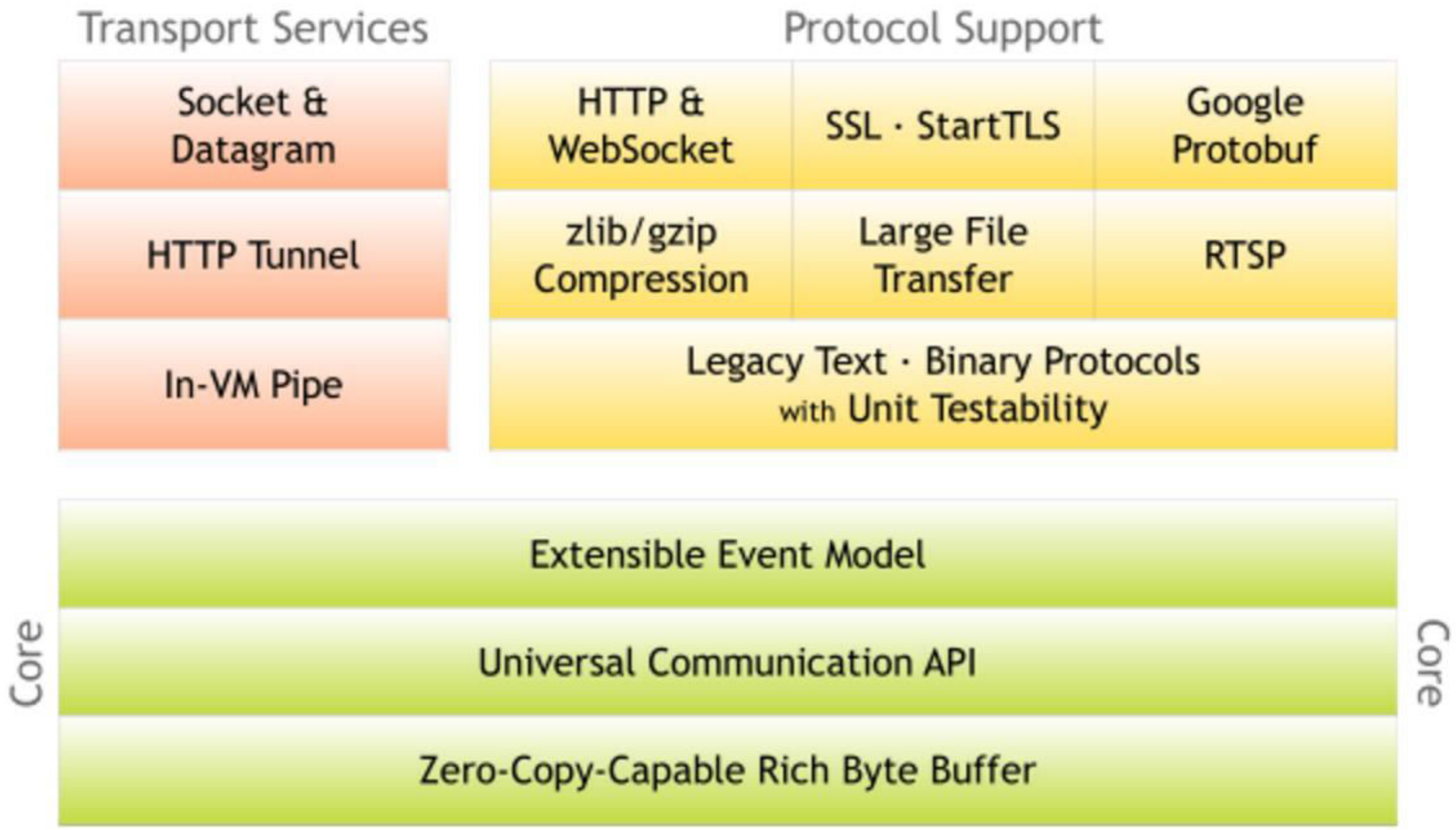
310 位联系人

Netty Features

- Unified API: BIO/NIO/AIO
- event model which allows clear separation of concerns
- Highly customizable thread model
- High performance & scalable
- Security:Complete SSL/TLS and StartTLS support
- Many known use case:

http://netty.io/wiki/adopters.html

Netty Arch



Transport Services

Socket & Datagram

HTTP Tunnel

In-VM Pipe

Protocol Support

HTTP & WebSocket

SSL · StartTLS

Google Protobuf

zlib/gzip Compression

Large File Transfer

RTSP

Legacy Text · Binary Protocols with Unit Testability

Core

Extensible Event Model

Universal Communication API

Zero-Copy-Capable Rich Byte Buffer

Core

Netty origin? Reactor Pattern

Douglas C. Schmidt

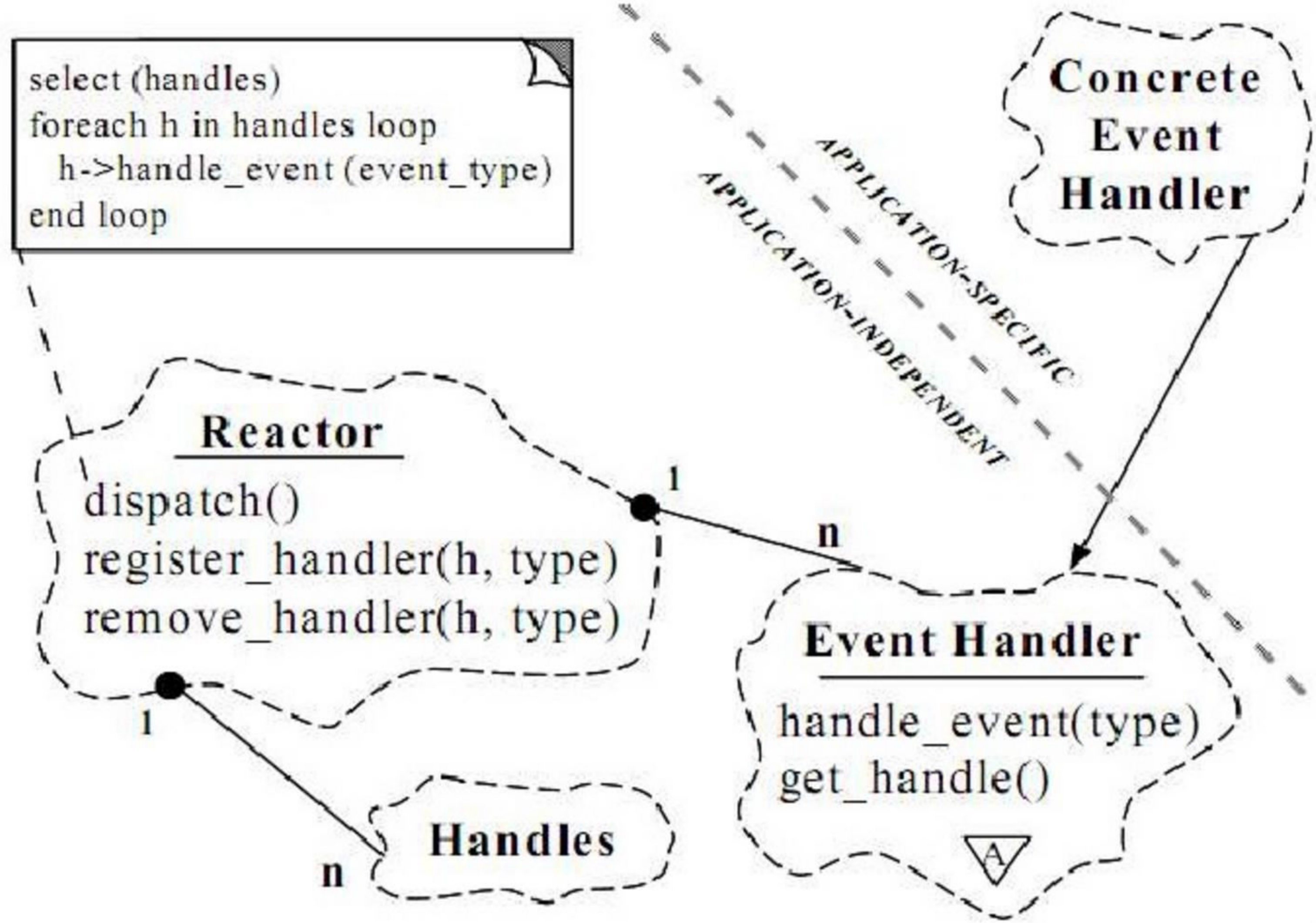
d.schmidt@vanderbilt.edu

Associate Chair of
Computer Science and
Engineering and Professor
of Computer Science,
at Vanderbilt University

1025, 16th Ave So.,
Nashville, TN 37212
Institute for Software
Integrated Systems (ISIS)
(615) 343-7472

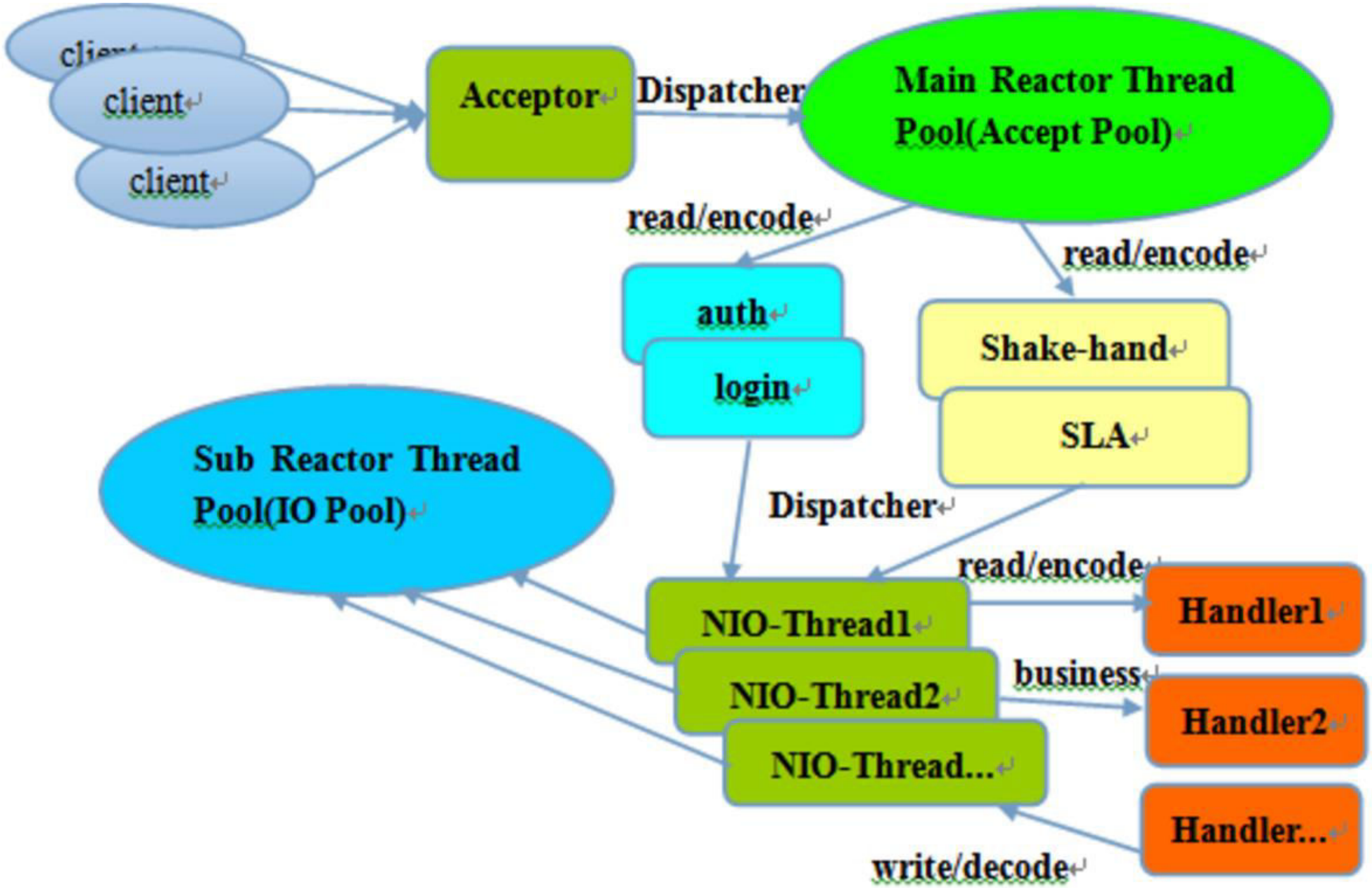
http://www.dre.vanderbilt.edu/~schmidt/
https://en.wikipedia.org/wiki/Douglas_C._Schmidt
http://www.cs.wustl.edu/~schmidt/PDF/reactor-siemens.pdf

Reactor Pattern

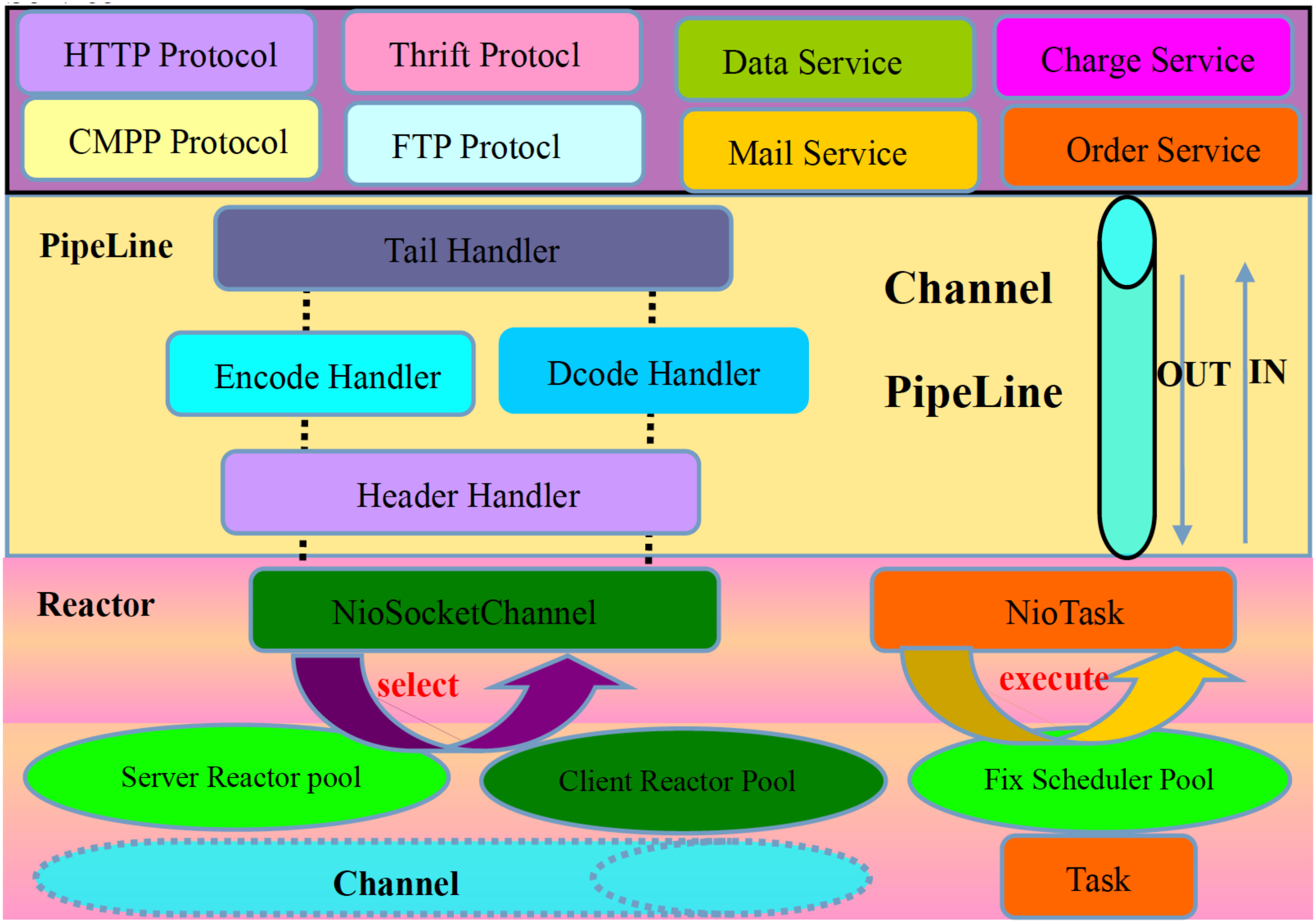


```
select (handles)
foreach h in handles loop
  h->handle_event (event_type)
end loop
```

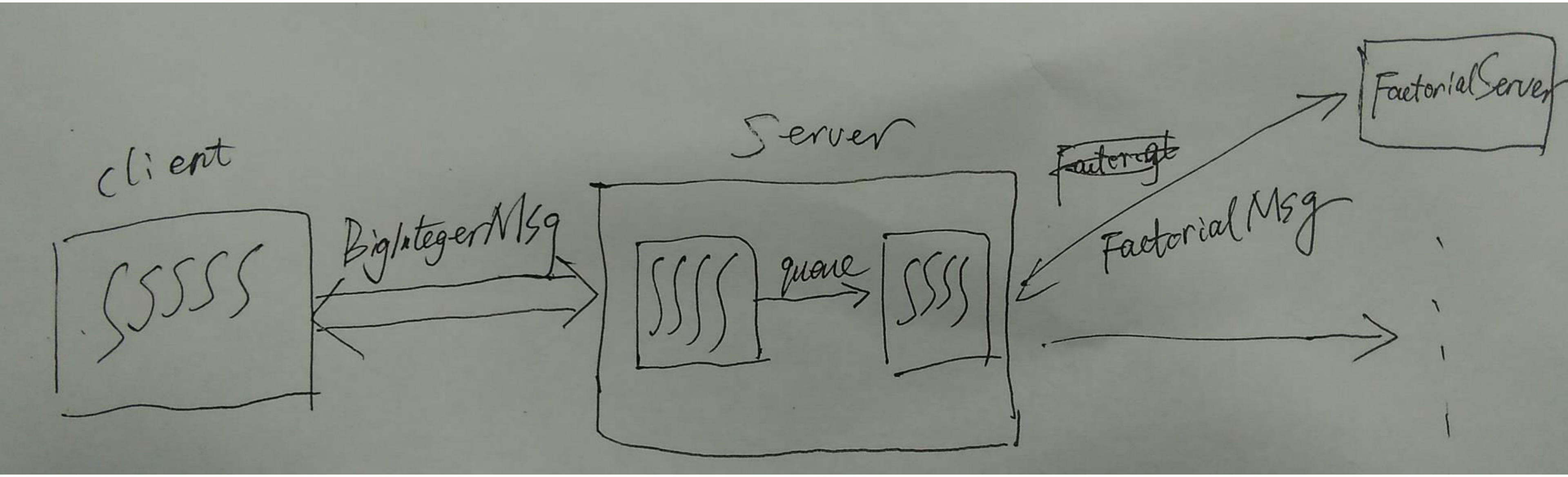
Netty thread model



Netty Layer


HTTP Protocol
Thrift Protocl
Data Service
Charge Service
CMPP Protocol
FTP Protocl
Mail Service
Order Service
PipeLine
Tail Handler
Channel
PipeLine
Encode Handler
Dcode Handler
OUT
IN
Header Handler
Reactor
NioSocketChannel
NioTask
select
execute
Server Reactor pool
Client Reactor Pool
Fix Scheduler Pool
Channel
Task

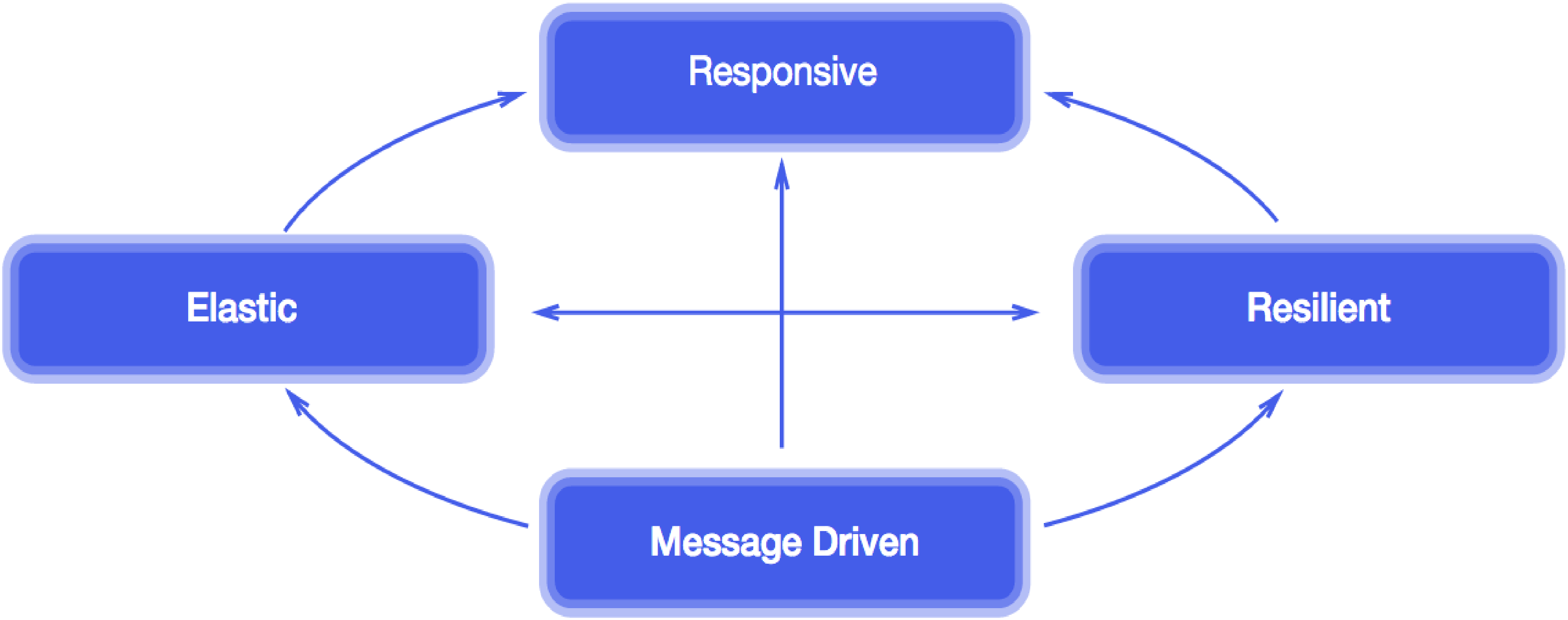
Netty demo



- ➢ Client simulate multiple clients which number can be configured.
- ➢ Server is Async and nonblock,
- ➢ FactorialServer is cpu bounded and can be horizon scalable

https://github.com/flyours/netty-drill

Reactive Web Application



http://www.reactivemanifesto.org/

reactive programs

continously available and readily responding to external demand.

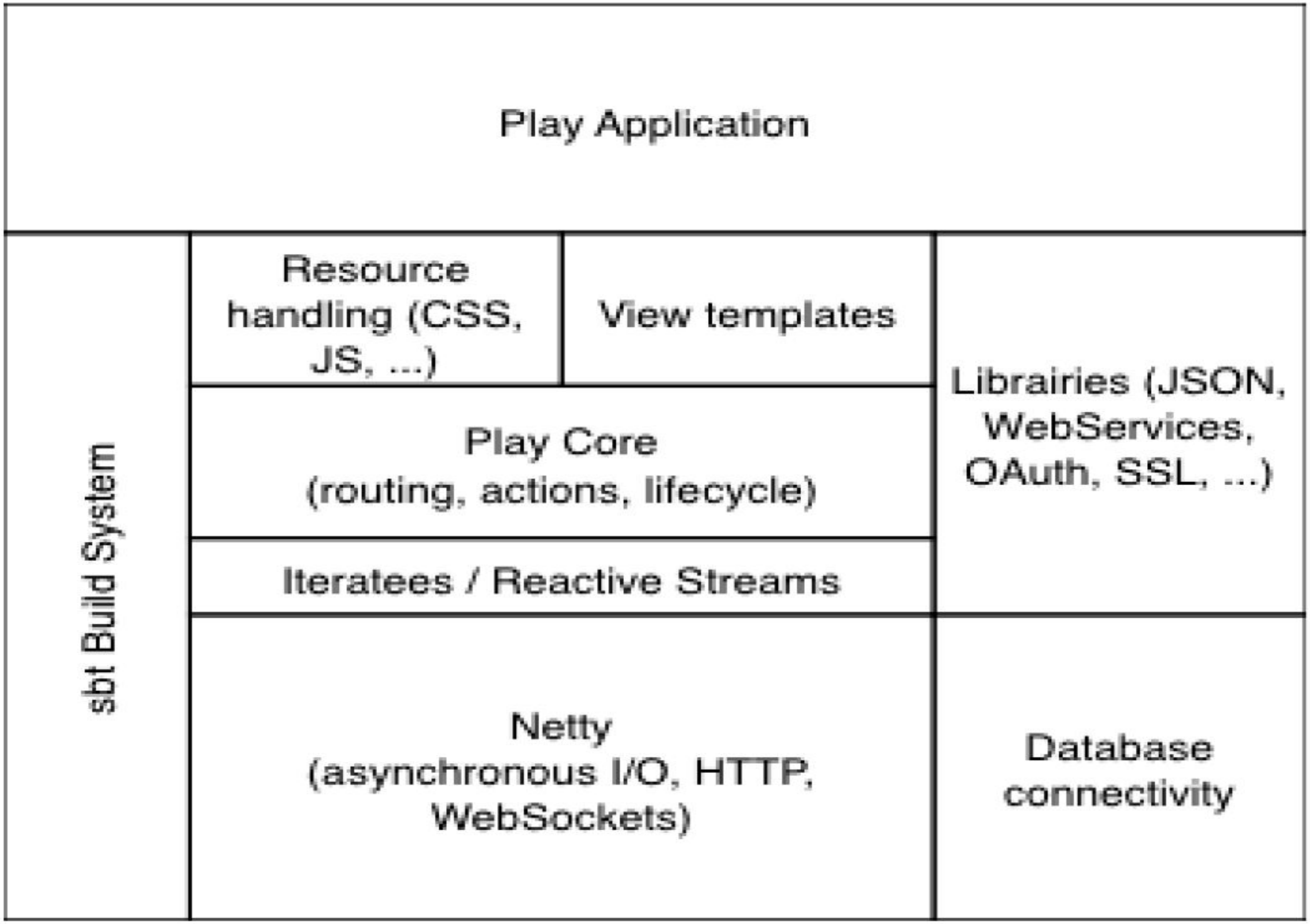
Prerequisites: asynchronous and event-driven

4 traits:

- responsive: react to users
- Scalable/Elastic: react to load
- resilient: react to failure
- event-driven/Messege-Driven: react to events

On JVM,the only mature full-stack reactive web-application framework to this day is the Play Framework

Play2 Arch



Play Application

sbt Build System

Resource handling (CSS, JS, ...)

View templates

Play Core (routing, actions, lifecycle)

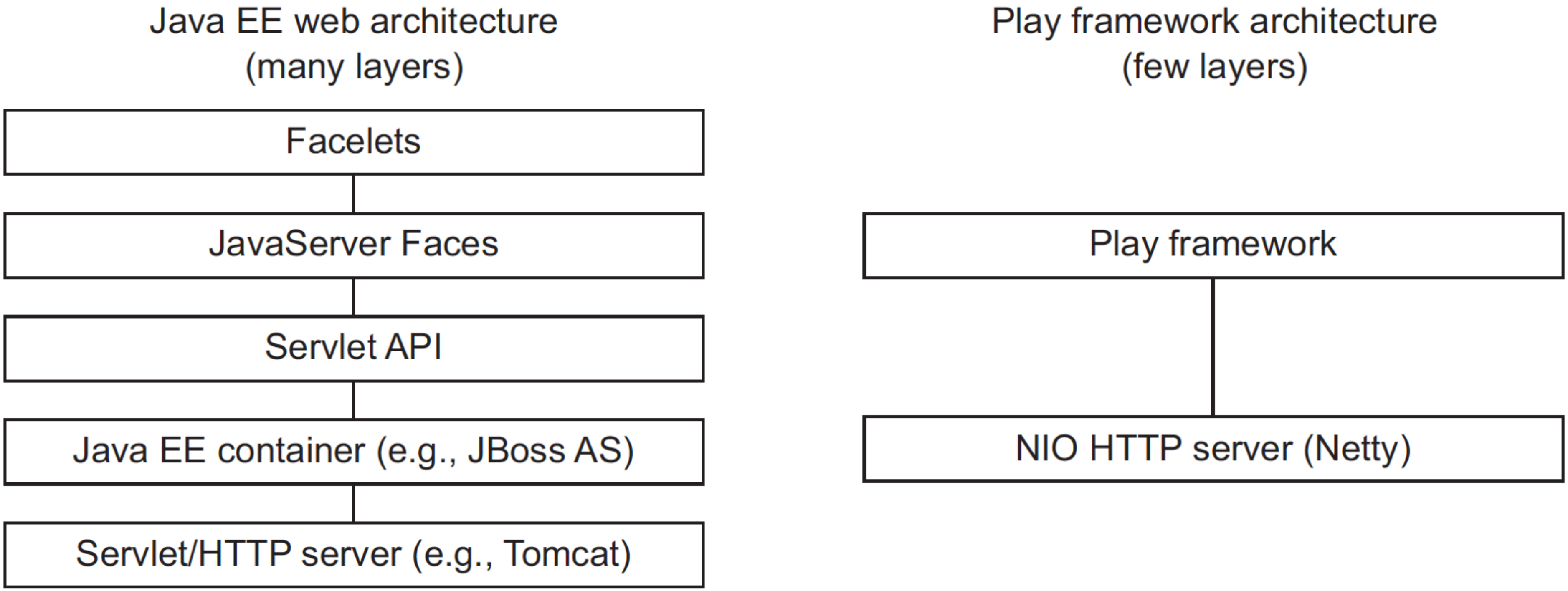
Iteratees / Reactive Streams

Netty (asynchronous I/O, HTTP, WebSockets)

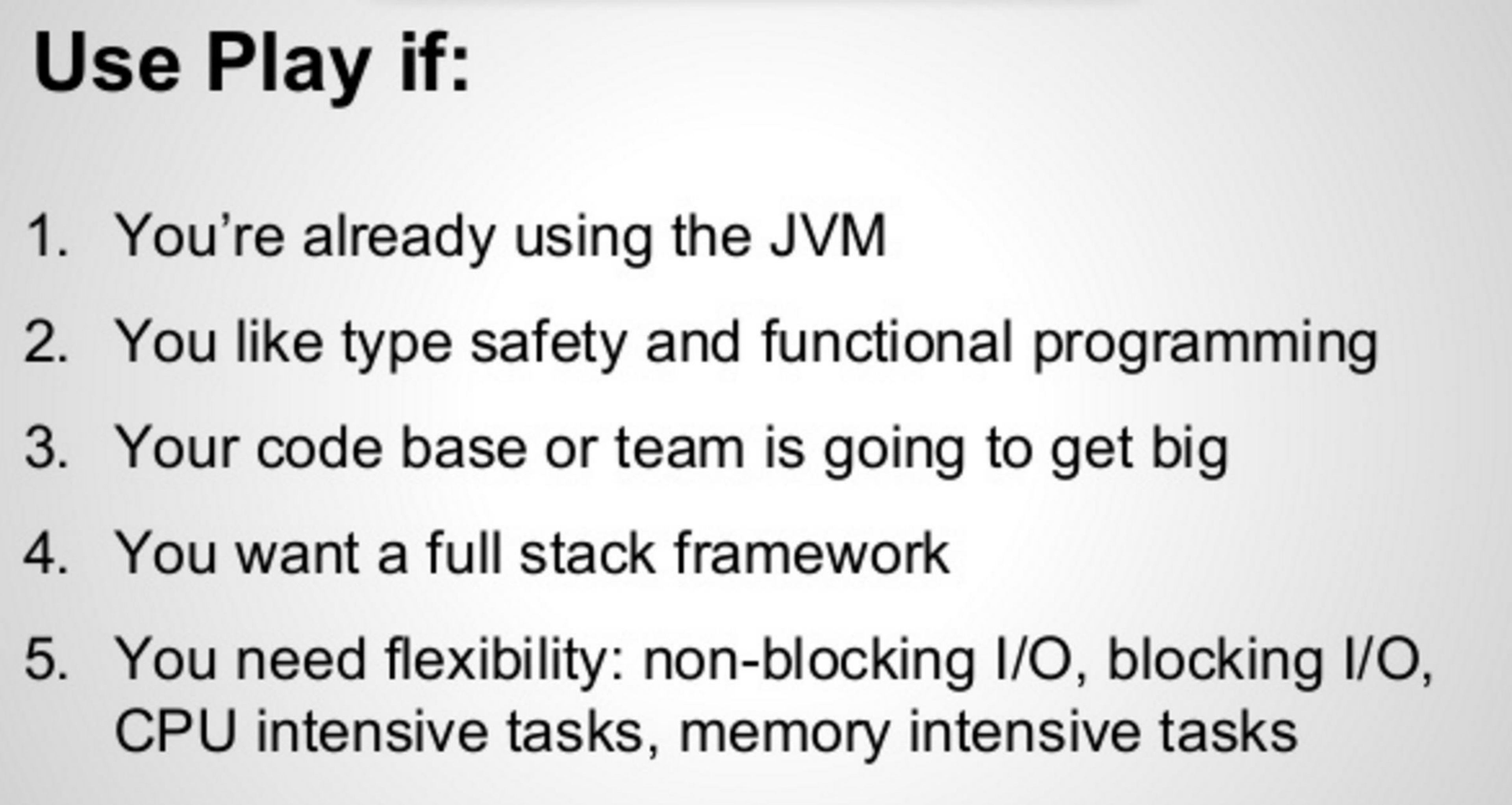
Librairies (JSON, WebServices, OAuth, SSL, ...)

Database connectivity

“lasagna” architecture VS simplified architecture



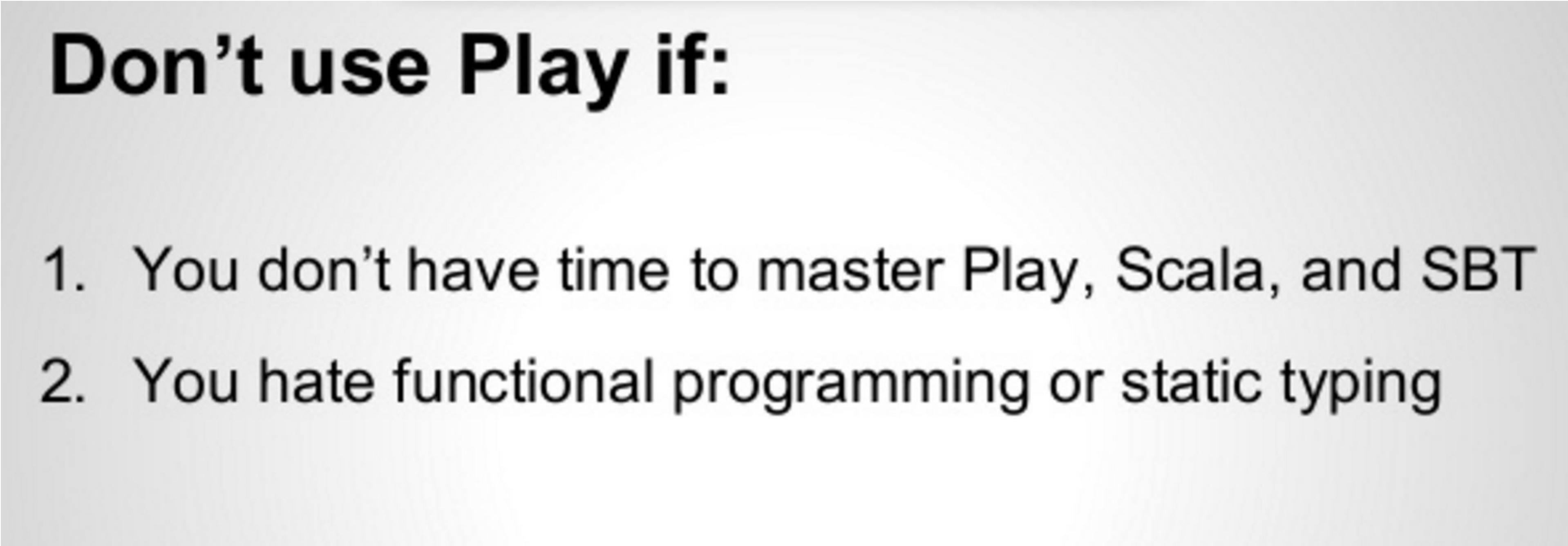
Why Play?



Use Play if:

1. You’re already using the JVM
2. You like type safety and functional programming
3. Your code base or team is going to get big
4. You want a full stack framework
5. You need flexibility: non-blocking I/O, blocking I/O, CPU intensive tasks, memory intensive tasks

Why not Play?



Don’t use Play if:

1. You don’t have time to master Play, Scala, and SBT
2. You hate functional programming or static typing

Play2 demo

- Activator new can be used to get scaffolding code.
- Full stack support: scala template/async actor

https://github.com/flyours/play2-drill

Reference

- http://netty.io/wiki/user-guide-for-5.x.html
- http://www.reactivemanifesto.org/
- http://www.reactive-streams.org/
- https://www.lightbend.com/activator/templates
- https://loicdescotte.github.io/posts/play25-akka-streams/