Software Product Development Experiences and Thoughts

倪文君 (Wen-Chun Ni)
叡揚資訊 (Galaxy Software Services)

About Me

- 叡揚資訊
  - 中央創新研究所 (CII) 所長
  - 架構辦公室負責人
- 台大資訊工程系所 (1985 and 1989)
- Brown University (1992)

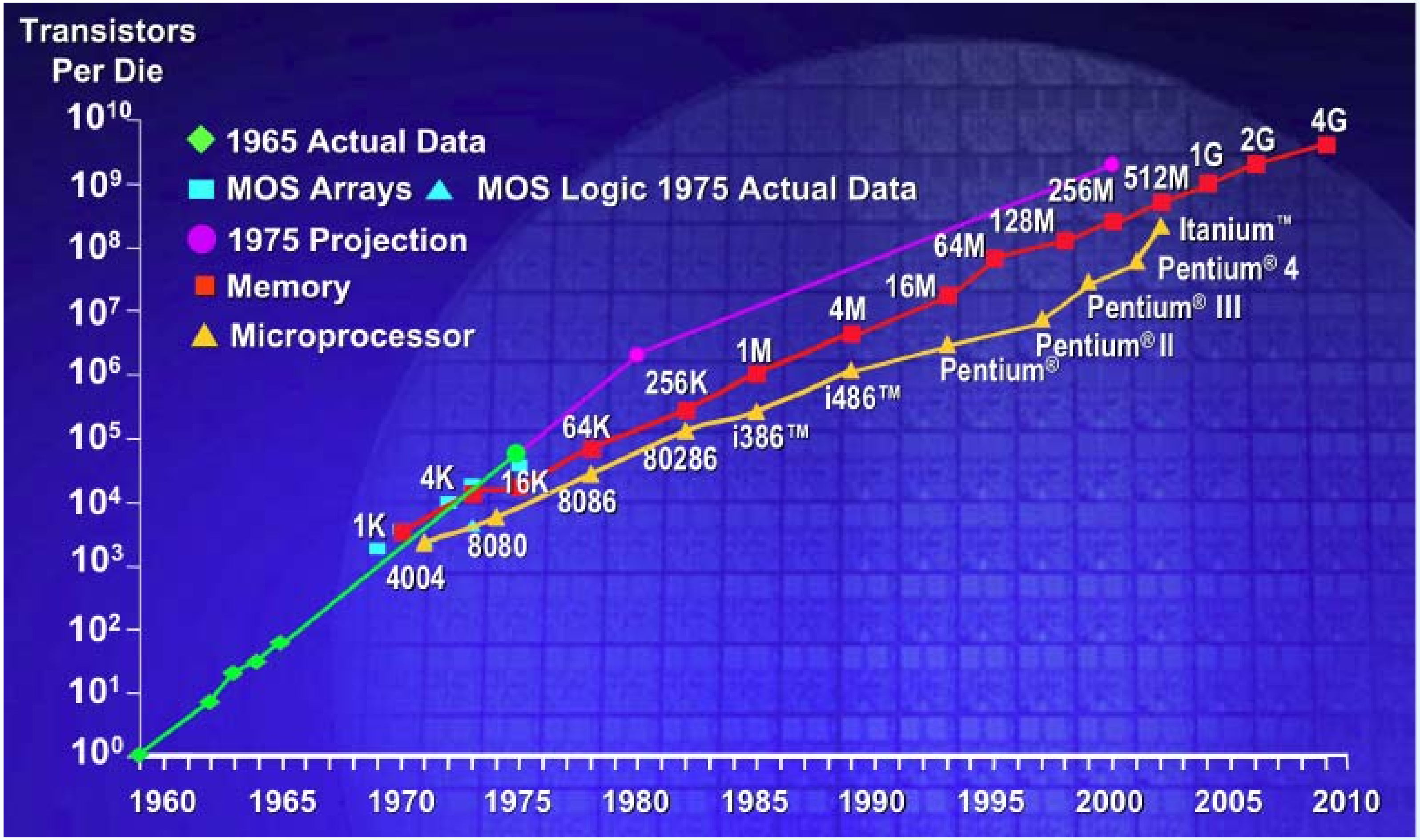
Our perpetual hope

The Computing Power

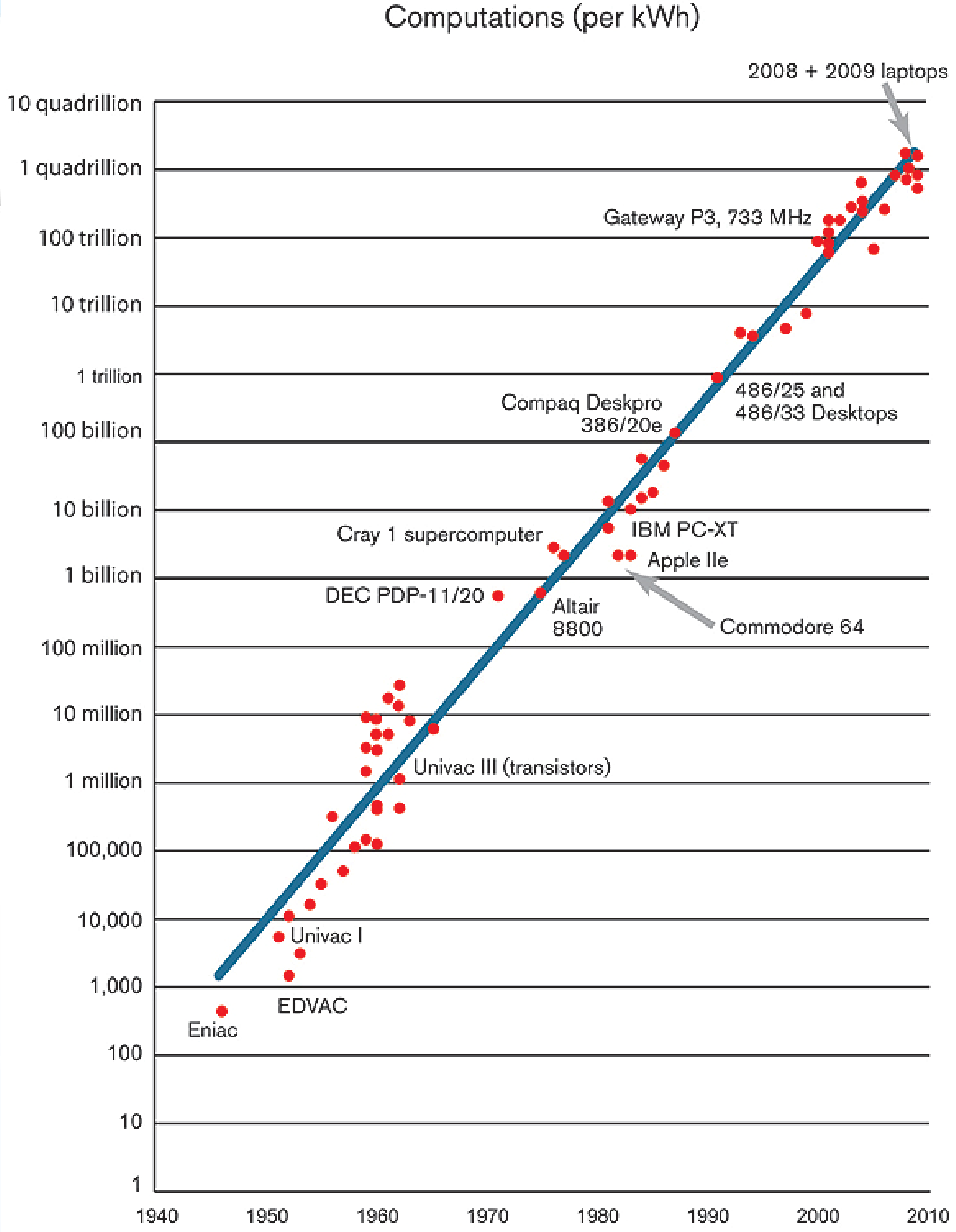
Moore’s Law

The number of transistors per unit doubles every two years (or 18 months).

Plot of Moore's Law



Moore’s Law:
The power version



The number of computation with same energy doubles every 18 months.

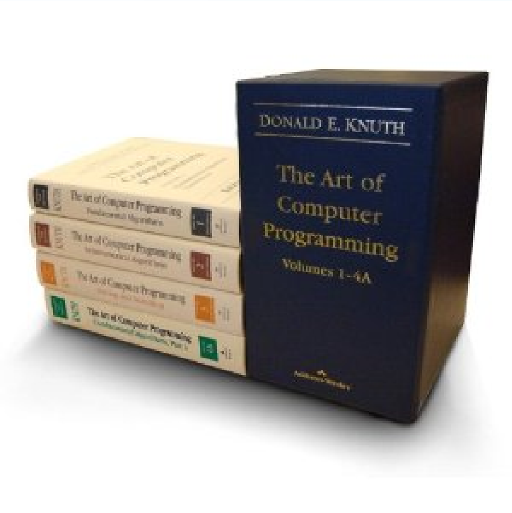
What’s wicked

The ~~Problem~~ Code Complexity

Software development is ***not*** hard, as long as you don’t have to change code.

Donald Knuth

[...] SOFTWARE IS HARD. From now on, I shall have significantly greater respect for every successful software tool that I encounter.



ANNE HAMERSKY

Sources of Complexity

- Requirement
- System Analysis and Design
- Code
- Validation and Verification
- Operations

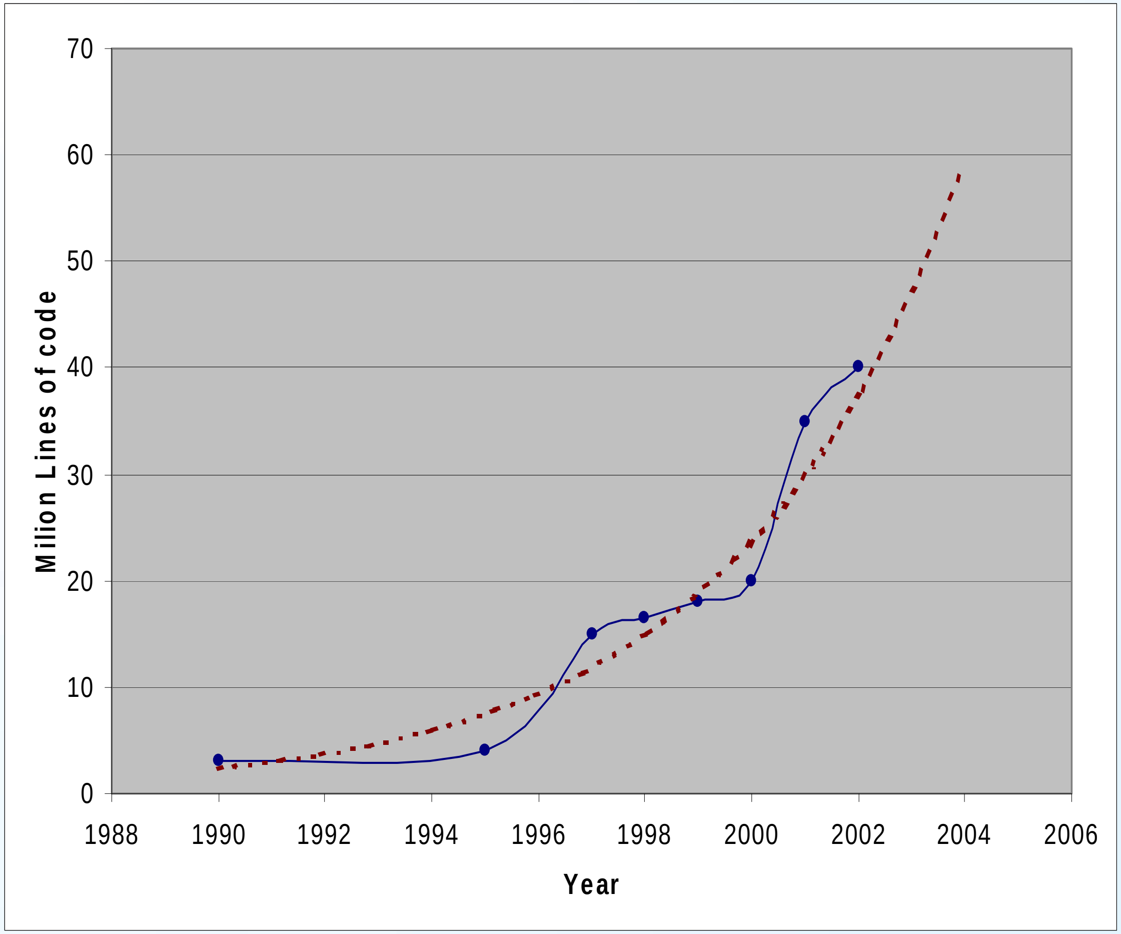
After several revisions, your code base will **bloat**.

Windows Operating Systems

Year	Operating System	SLOC (Million)
1993	Windows NT 3.1	4-5
1994	Windows NT 3.5	7-8
1996	Windows NT 4.0	11-12
2000	Windows 2000	more than 29
2001	Windows XP	45
2003	Windows Server 2003	50

A complexity measure of Windows family

1990	Win 3.1
1995	Win NT
1997	Win 95
1998	NT 4.0
1999	Win98
2000	NT 5.0
2001	Win2k
2002	XP



Non-Windows Operating Systems

Operating System	SLOC (Million)
Debian 2.2	55-59
Debian 3.0	104
Debian 3.1	215
Debian 4.0	283
Debian 5.0	324
OpenSolaris	9.7
FreeBSD	8.8
Mac OS X 10.4	86
Linux kernel 2.6.0	5.2
Linux kernel 2.6.29	11.0
Linux kernel 2.6.32	12.6
Linux kernel 2.6.35	13.5
Linux kernel 3.6	15.9
Linux kernel pre-4.2	20.2

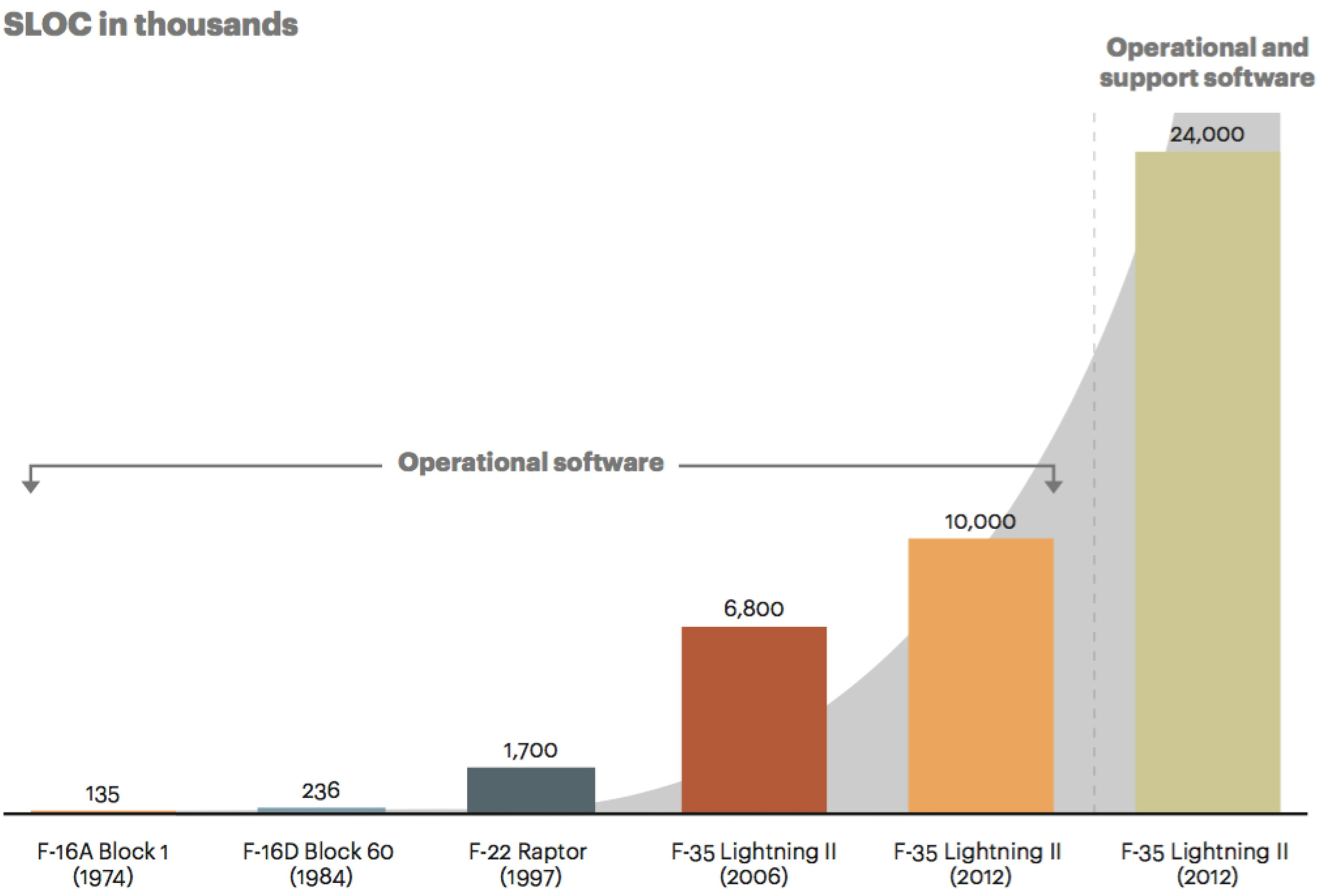
Size Comparisons of Embedded Software

System	Lines of Code	Language
Mars Reconnaissance Orbiter	545K	C
F-22 Raptor	2.5M	Ada (90%)
Seawolf Submarine Combat System AN/BSY-2	3.6M	Ada
Boeing 777	4M	Ada
Boeing 787	7M	Ada (largely)
F-35 Joint Strike Fighter	19M	C and C++
Typical GM car in 2010	100M	MISRA-C for critical systems

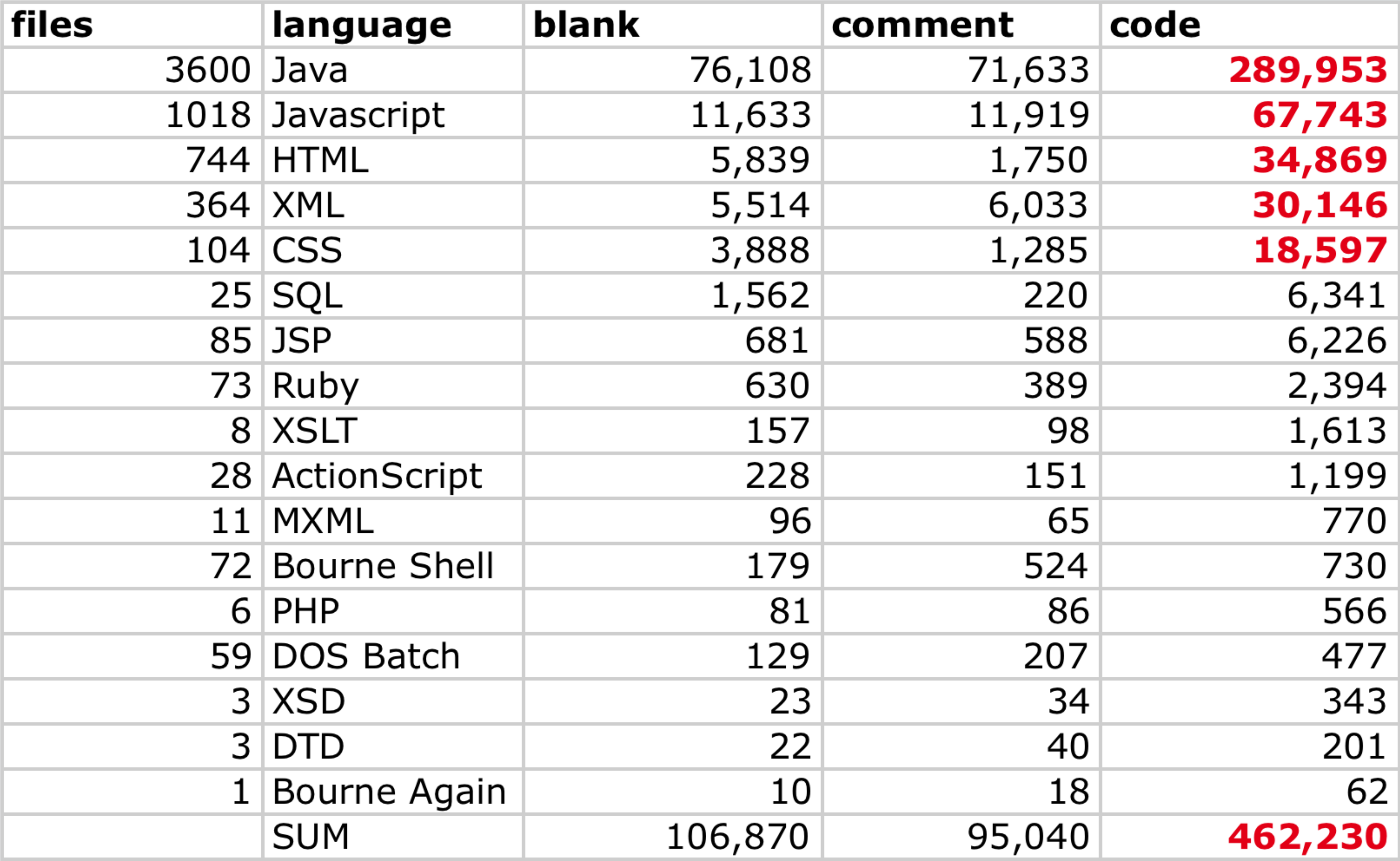
Modern Fighters’ Software

Figure 1

The number of source lines of code (SLOC) has exploded in avionics software



The Complexity of teamKube 2011-06-30



files	language	blank	comment	code
3600	Java	76,108	71,633	**289,953**
1018	Javascript	11,633	11,919	**67,743**
744	HTML	5,839	1,750	**34,869**
364	XML	5,514	6,033	**30,146**
104	CSS	3,888	1,285	**18,597**
25	SQL	1,562	220	6,341
85	JSP	681	588	6,226
73	Ruby	630	389	2,394
8	XSLT	157	98	1,613
28	ActionScript	228	151	1,199
11	MXML	96	65	770
72	Bourne Shell	179	524	730
6	PHP	81	86	566
59	DOS Batch	129	207	477
3	XSD	23	34	343
3	DTD	22	40	201
1	Bourne Again	10	18	62
	SUM	106,870	95,040	**462,230**

The Complexity of teamKube 2012-05-31

files	language	blank	comment	code
4,583	Java	100,417	94,170	377,050
2,283	Javascript	18,869	15,652	150,377
1,034	HTML	6,924	6,047	67,911
509	XML	6,958	5,951	40,353
146	CSS	4,417	1,685	22,953
128	JSP	1,303	989	9,013
29	SQL	1,580	236	6,504
10	XSD	64	63	2,881
16	XSLT	185	106	2,486
73	Ruby	630	389	2,394
28	ActionScri	228	151	1,199
11	MXML	96	65	770
82	Bourne Sh	142	431	606
6	PHP	81	86	566
7	Groovy	156	101	484
68	DOS Batch	118	181	415
4	DTD	59	76	294
1	Bourne Ag	11	18	68
9,018	SUM	142,238	126,397	686,324

The Complexity of teamKube 2013-05-27

files	language	blank	comment	code
5,676	Java	153,149	162,247	572,943
2,416	Javascript	30,883	35,143	233,403
1,236	HTML	6,863	5,939	73,012
611	XML	7,075	5,842	49,047
178	CSS	6,517	2,976	42,262
180	JSP	2,435	1,805	12,861
33	SQL	1,616	236	6,941
13	XSD	116	82	3,202
18	XSLT	178	112	2,993
73	Ruby	630	389	2,394
62	ActionScri	859	1,229	3,849
16	MXML	161	125	943
106	Bourne Sh	143	437	688
6	PHP	81	86	566
25	Groovy	423	190	2,155
92	DOS Batch	131	222	584
6	DTD	209	616	430
2	Bourne Ag	23	42	136
10,832	SUM	211,492	217,718	1,008,409

The Complexity of teamKube 2015-12-10

Language	files	blank	comment	code
Javascript	3466	100381	111788	785863
Java	7524	198193	200278	754349
C/C++ Header	175	4054	6751	453037
XML	1377	11644	10069	163521
HTML	1688	10911	6733	122357
CSS	360	16358	5400	99211
JSP	319	10101	4376	38248
Objective C	82	4068	4358	17349
Ant	123	2497	1960	14054
C++	20	2196	3170	10752
Velocity Template Language	302	957	50	9617
C#	28	1618	2097	9201
SASS	27	835	192	8882
SQL	37	1617	236	7188
Groovy	112	1012	614	5091
JSON	106	11	0	4826
XSD	22	165	108	4328
ActionScript	63	861	1229	3859
XSLT	19	209	128	3467
Ruby	75	641	414	2411
C	1	406	60	2193
Bourne Shell	184	236	666	2085
D	14	0	0	1998
DOS Batch	172	288	459	1391
Maven	7	75	115	1143
Bourne Again Shell	21	223	493	937
MXML	16	161	160	908
diff	1	0	97	876
NAnt script	2	18	0	702
PHP	8	88	100	620
DTD	7	215	649	568
Objective C++	1	168	123	567
QML	4	11	66	362
Swift	1	30	14	226
XAML	3	23	63	94
MSBuild script	1	0	7	82
Python	2	17	15	49
ASP.Net	4	0	0	36
Prolog	1	2	0	15
YAML	1	0	0	5
SUM:	16376	370290	363038	2532468

Afterthoughts

- Annual Growth in Lines of Code: 50%
  - Complexity grows *exponentially*
- Polyglot and multiple platforms
- Mobile portability:
  - Sudden surge of JavaScript
  - Sudden surge of C/C++

Mobile Influence

Cross Mobile Platform

PhoneGap Plugin

1. JNI
2. Objective-C++
3. Windows RT Component in C++

Bridge JavaScript & C++

Cross Mobile & Desktop Platform

Grove giveth and Gates taketh away.

-- Bob Metcalfe

Complex Data

- Unstructured: Microsoft Office files
- Semi-structured: XML, HL7
- Complex structured: Hierarchical XML

teamKube's Elements

Places

Documents

Activities

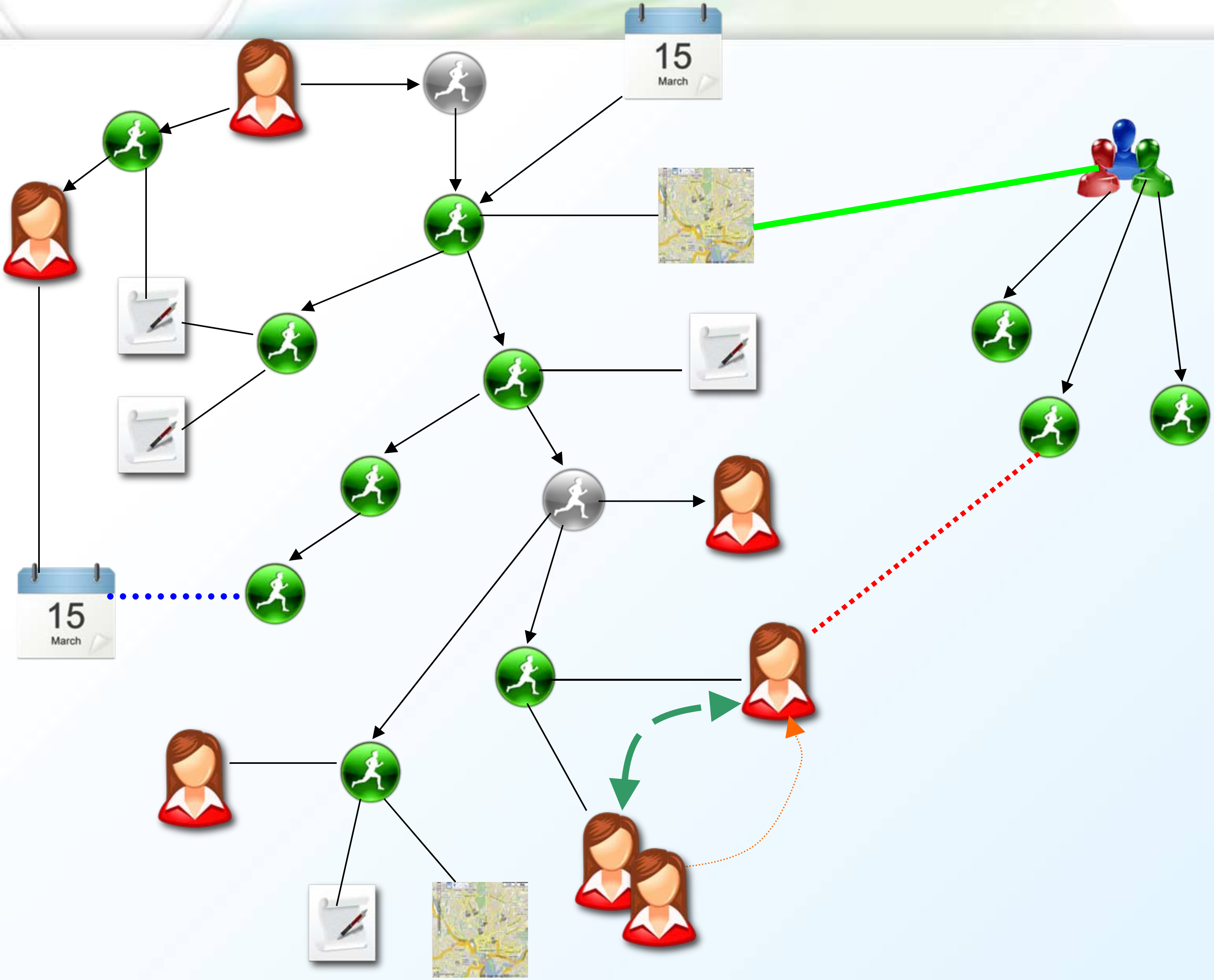
Tasks

Groups

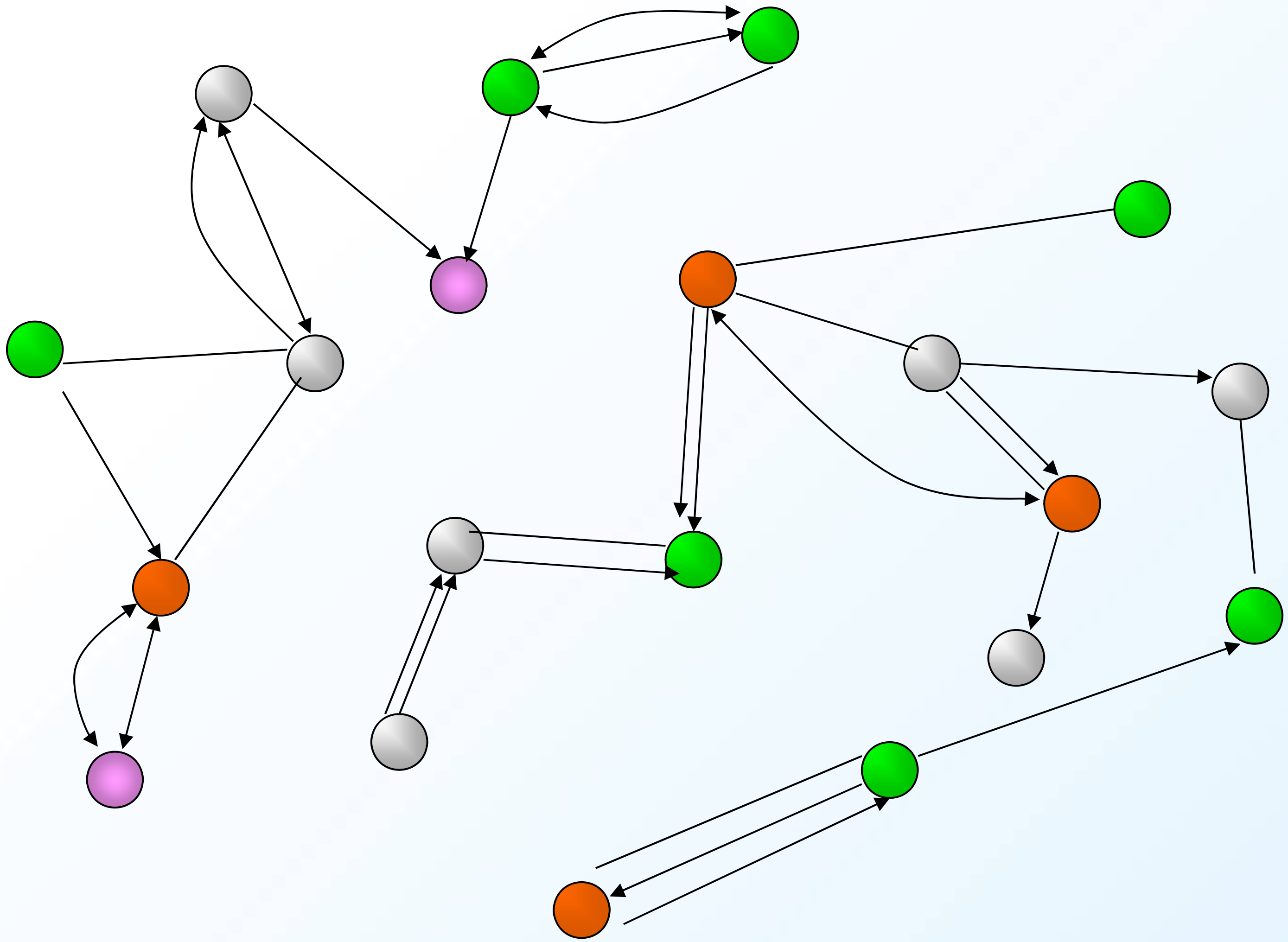
Person

Time

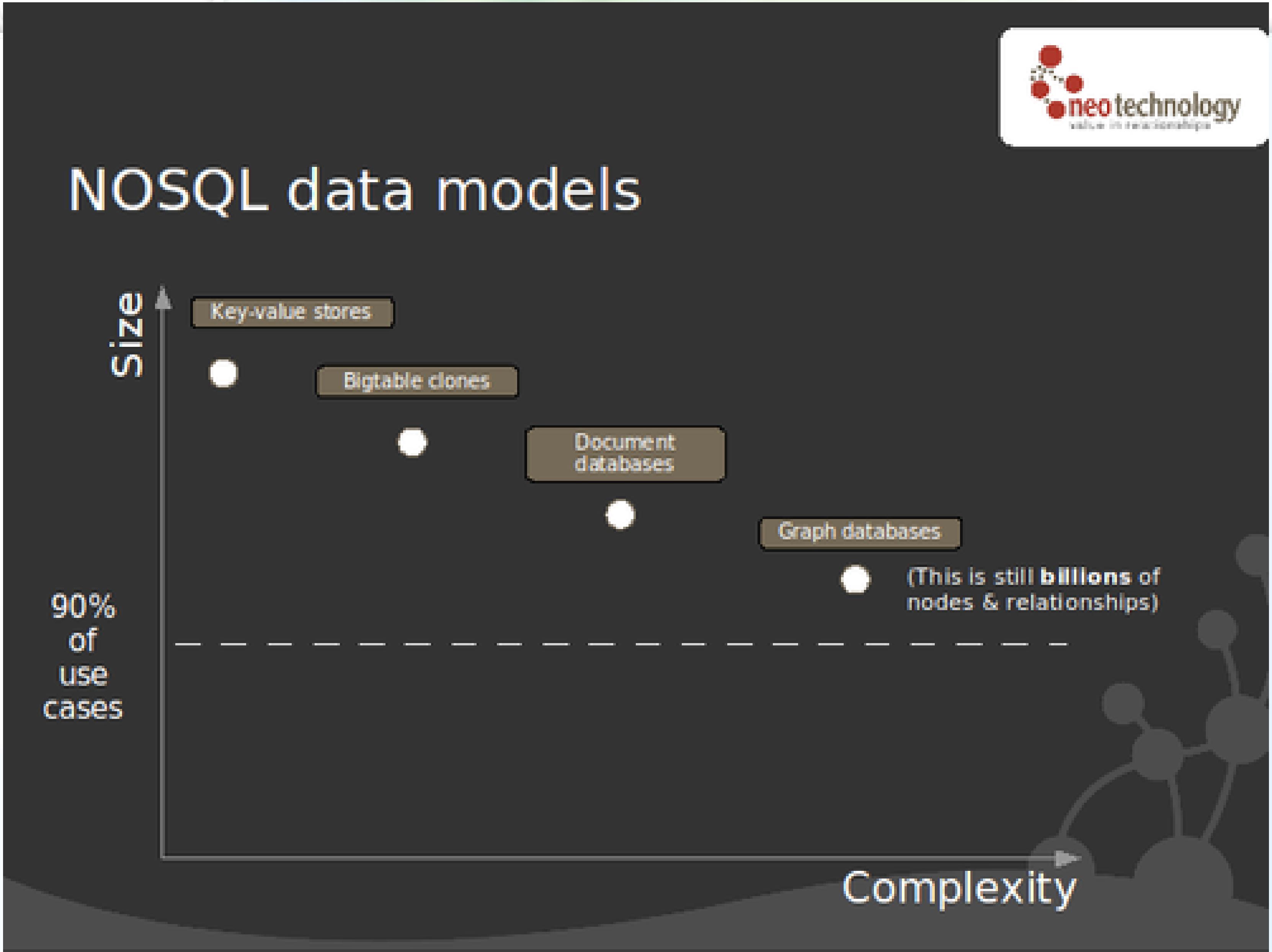
Data in teamKube



Abstract form: multi-graph



The Next Frontier of teamKube



NOSQL data models

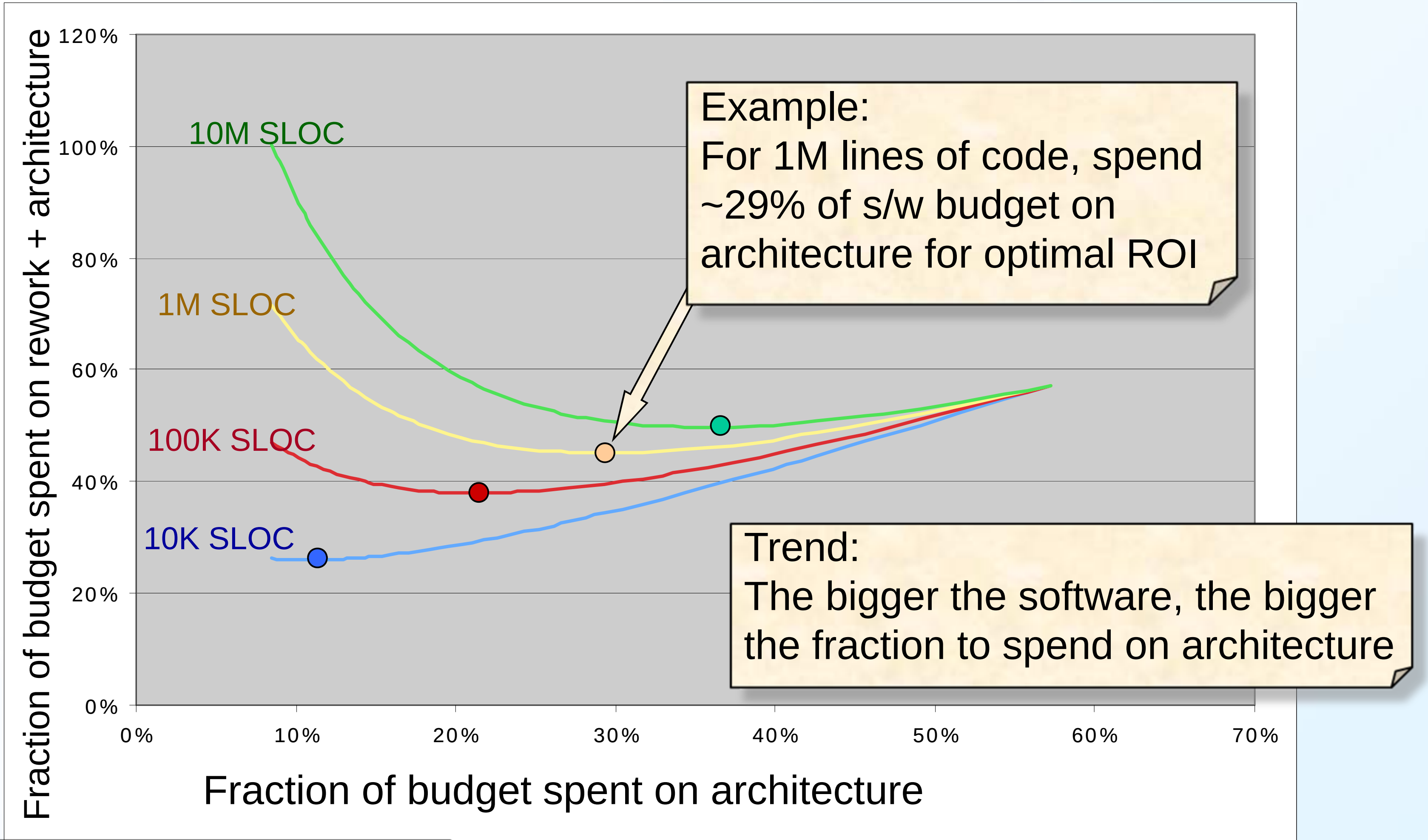
How we set design constraints (and services)

The Architecture

Architecture is the set of early decisions that are extremely costly to change later.

Architecture Investment “Sweet Spot”

Predictions from COCOMO II model for software cost estimation



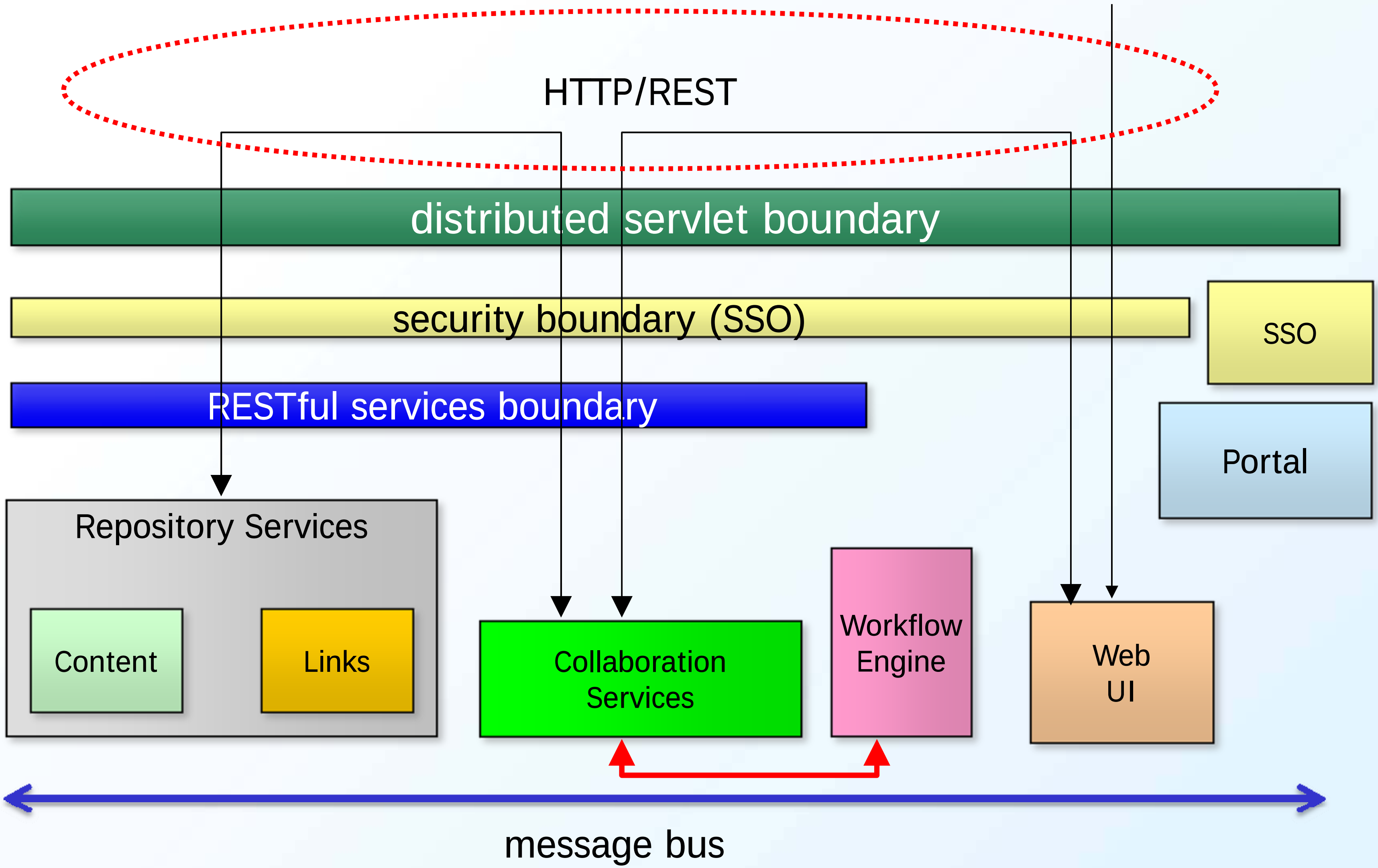
Example:
For 1M lines of code, spend ~29% of s/w budget on architecture for optimal ROI

Trend:
The bigger the software, the bigger the fraction to spend on architecture

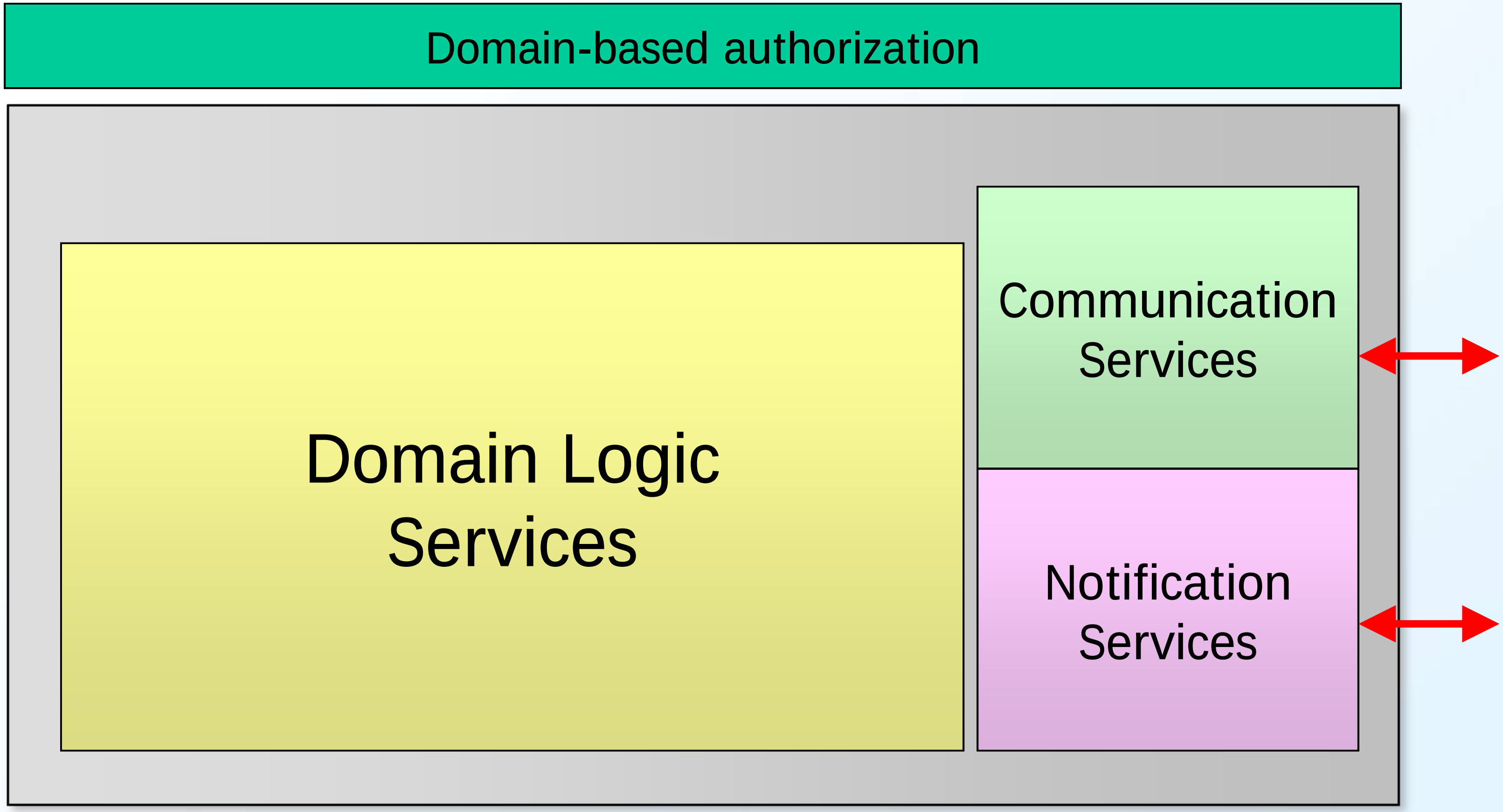
Note:
Prior investment in a reference architecture pays dividends

Source: Kirk Reinholtz, JPL

teamKube's Architecture



Collaboration Services



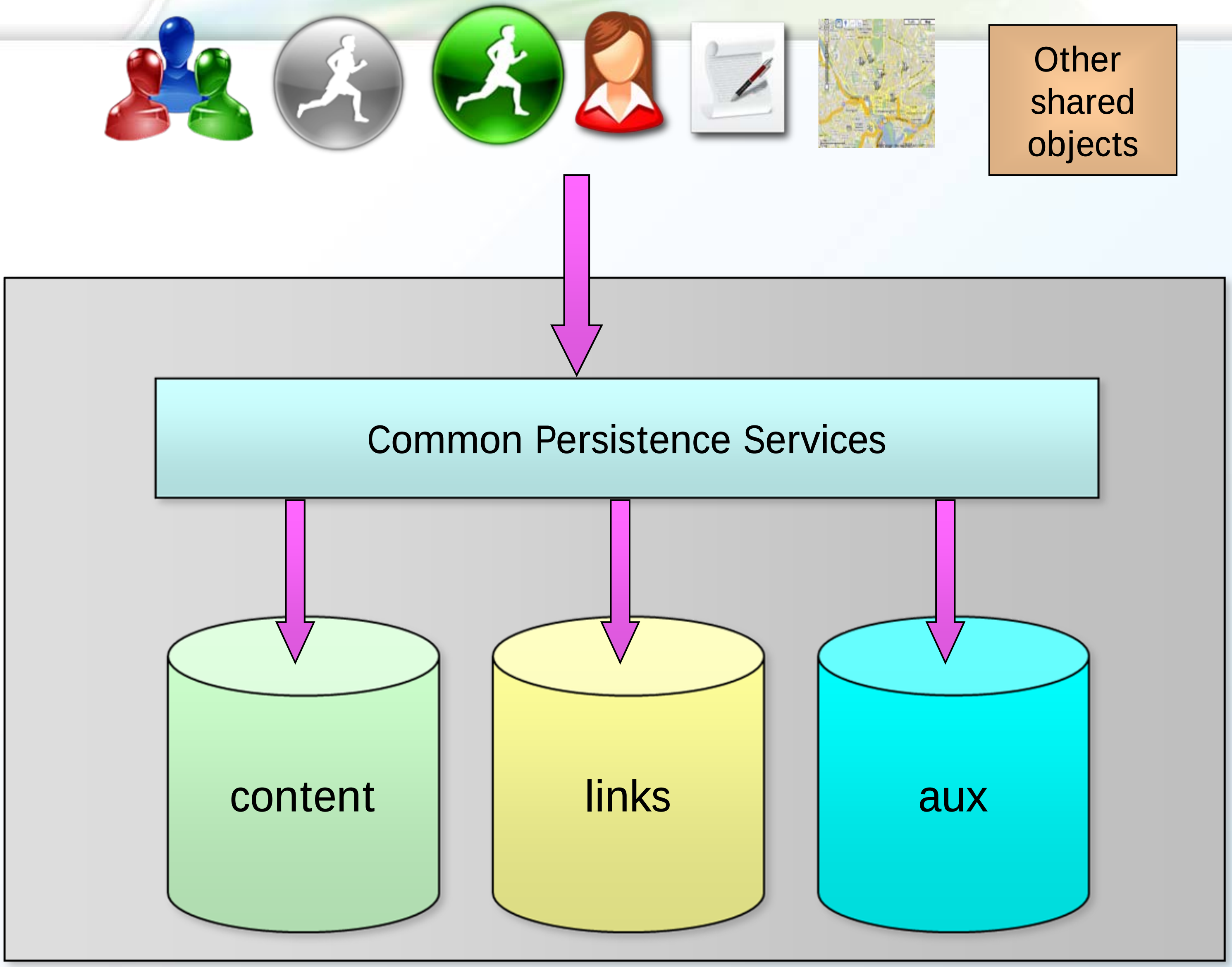
Domain-based authorization

Domain Logic Services

Communication Services

Notification Services

Repository Services



Other shared objects

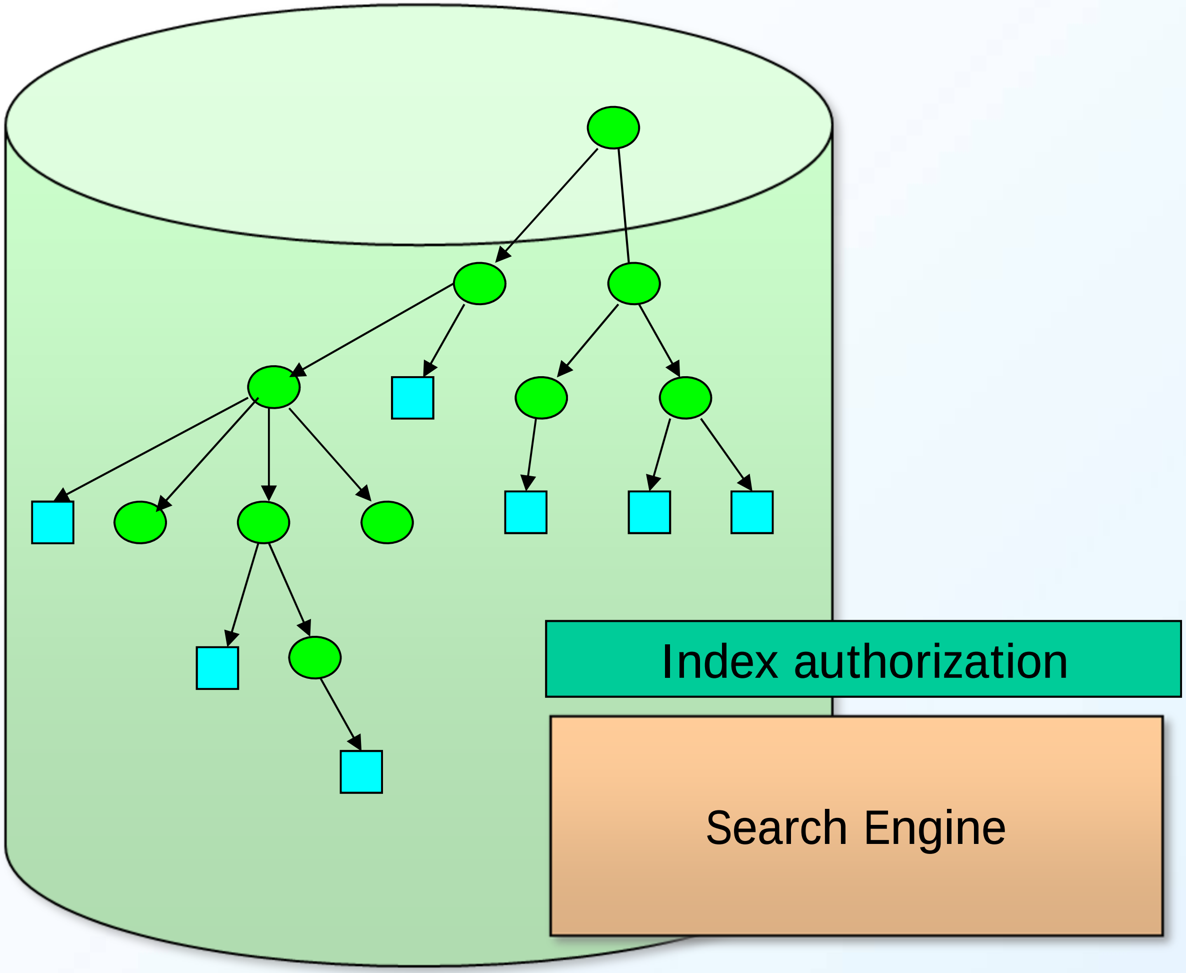
Common Persistence Services

content

links

aux

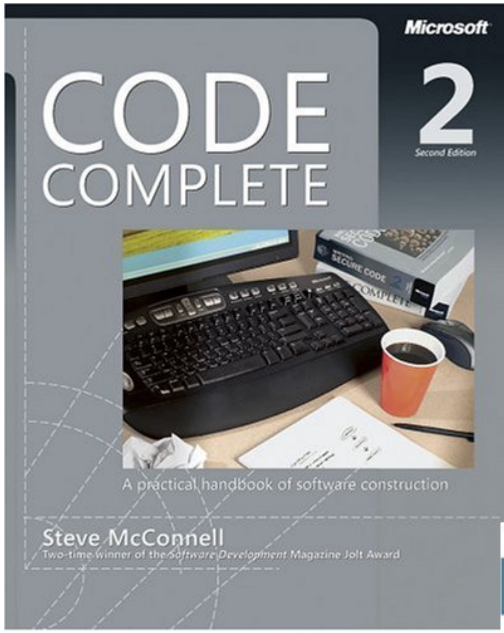
Content Service

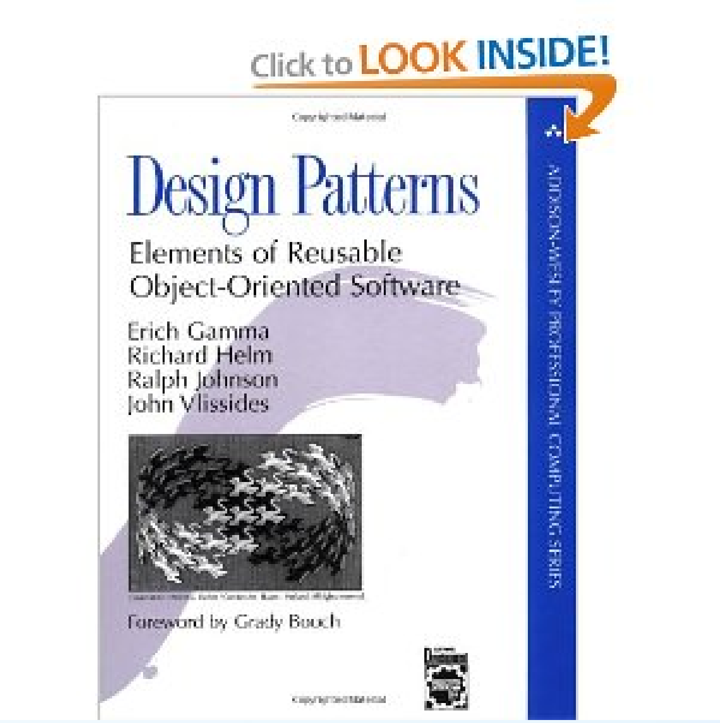


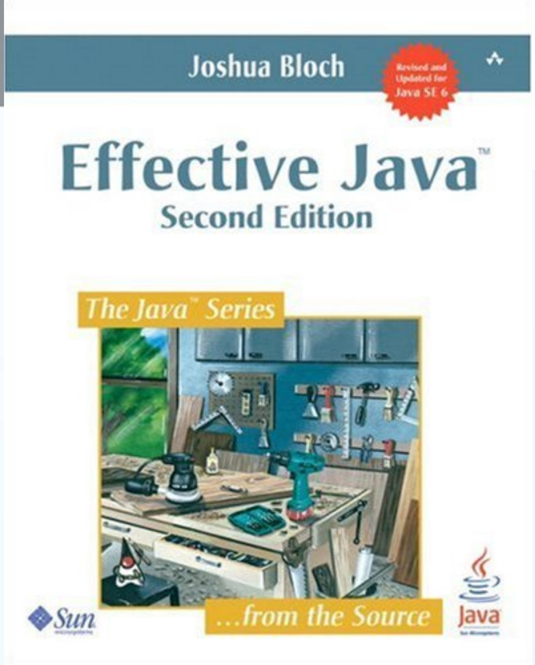
Team Learning

- Reviews
  - Analysis
  - Design
  - Code
- Basic Readings
- Advanced Readings
- *What's New* Sessions

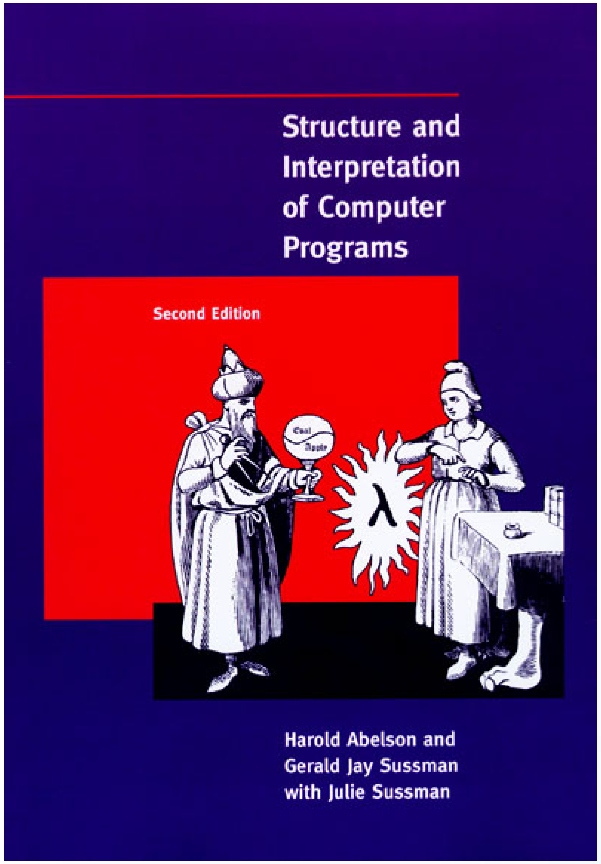
Basic Readings

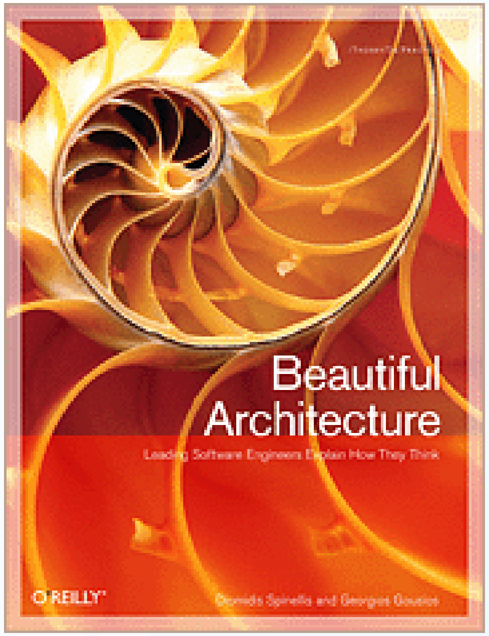

Microsoft
CODE COMPLETE
2
Second Edition
A practical handbook of software construction
Steve McConnell
Two-time winner of the Software Development Magazine Jolt Award

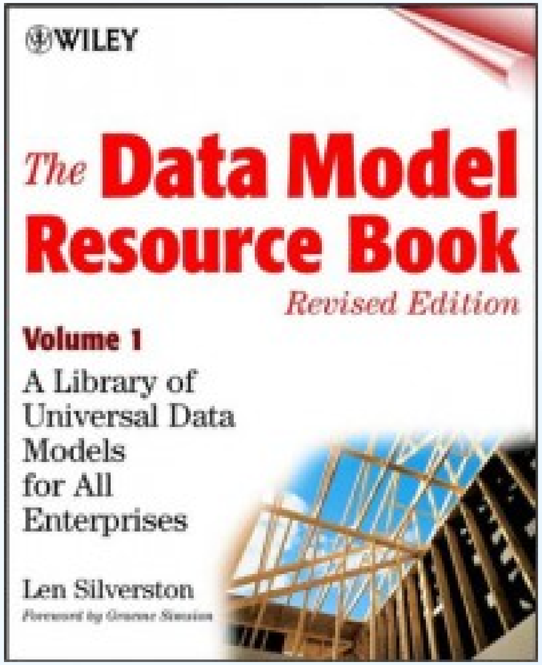




Advanced Readings


Structure and Interpretation of Computer Programs
Second Edition
λ
Harold Abelson and Gerald Jay Sussman with Julie Sussman


Theory in Practice
Beautiful Architecture
Leading Software Engineers Explain How They Think
O'REILLY
Diomidis Spinellis and Georgios Gousios


WILEY
The Data Model Resource Book
Revised Edition
Volume 1
A Library of Universal Data Models for All Enterprises
Len Silverston
Foreword by Graeme Simsion

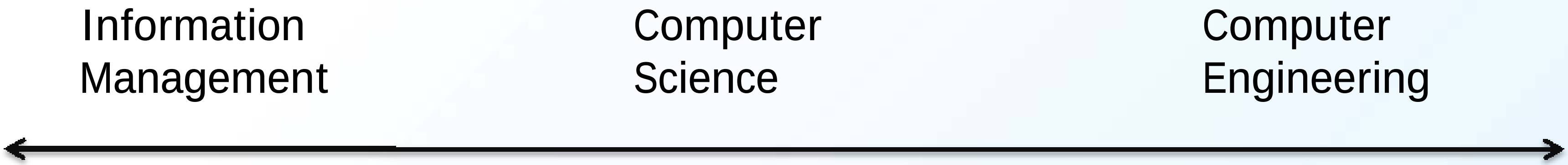
A reflection on education

Does Computer Science Matter?

Why a CS Degree Is Dismissed?

- Computer Science != A Science about Computers
- Computer scientist != Software engineer
- Why no certificate of software engineers?
  - The field is fast moving
  - The definition of computer changes
  - It's based on merits, not academic or other credentials
  - Most college CS graduates suck at CS
- Hiring software engineer – the requirement?
  - Only 'programming experiences'

The Spectrum



Information Management

Computer Science

Computer Engineering

Information Management

- Databases and applications
- Business Applications
  - CRM (Customer Relationship Management)
  - ERP (Enterprise Resource Planning)
  - HCM (Human Capital Management)
  - Financial Software
  - Office Automation

Computer Engineering

- Computer Architecture and Organization
- VLSI
- Networks
- Sensor technologies
- Computer and Communication

The Core of Computer Science

- Algorithms
- Discrete Math
- Theory of Computation
- Programming Languages
- Software Design Principles
- System Software
  - Compiler
  - Operating Systems
  - Concurrent and Distributed Computing

Computer Science and teamKube

- Predicate logic and set theory: rules
- Graph theory and algorithms: repository
- Automata theory: object states
- Compiler – parser: DSL
- Concurrency and asynchrony: scaling
- Software design
  - Object-oriented programming
  - Functional programming

To Educators

- Holistic Approach to Future Courses
  - Inject software in *every* course
- Programming languages exposure:
  - Different programming paradigms
  - Lisp is useful
- Heavier and bigger programming projects
- Deep fundamental knowledge will pay back

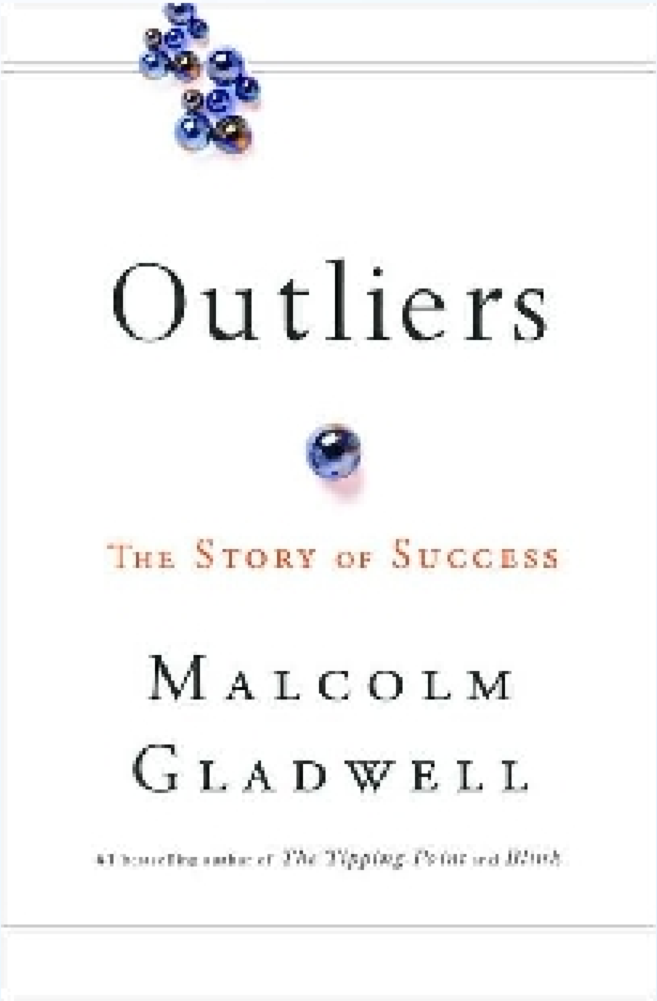
How to become good at it

The Efforts

"The illiterate of the 21st Century will not be those who cannot read or write, but those who cannot learn, unlearn and relearn.“

-- Alvin Toffler

It takes **10,000** hours



Outliers

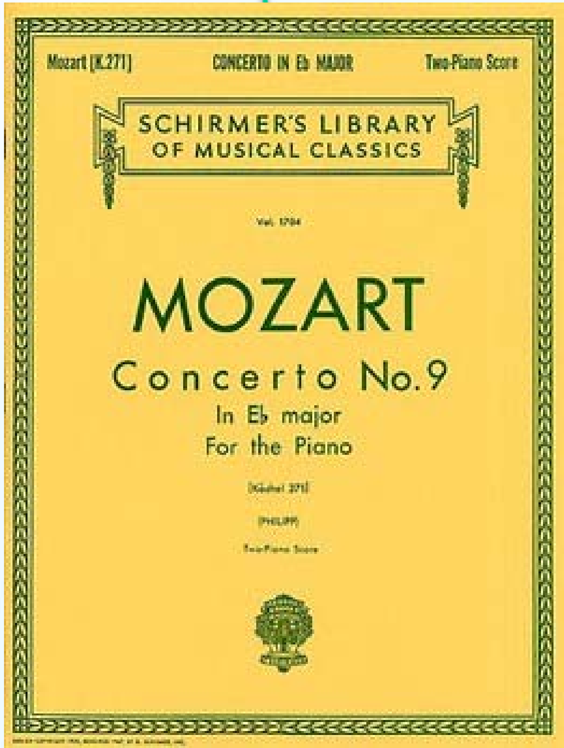
THE STORY OF SUCCESS

MALCOLM GLADWELL

#1 bestselling author of *The Tipping Point* and *Blink*

Mozart

6 years old



Mozart [K.271] CONCERTO IN E♭ MAJOR Two-Piano Score

SCHIRMER'S LIBRARY
OF MUSICAL CLASSICS

Vol. 1784

MOZART

Concerto No. 9
In E♭ major
For the Piano

[Köchel 271]

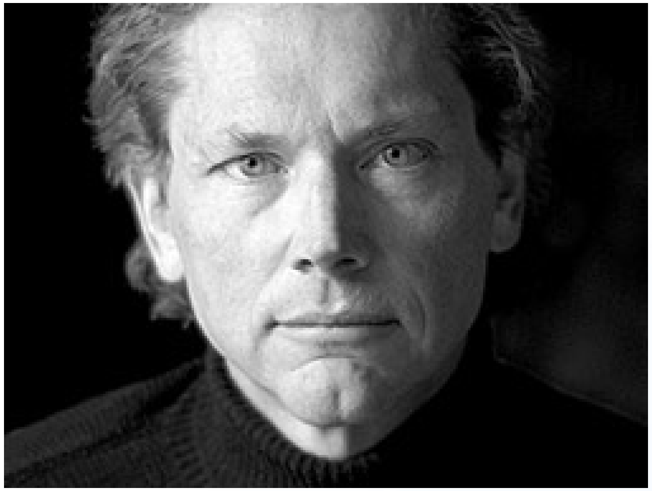
(PHILIPP)

Two-Piano Score

21 years old

The Beatles in Hamburg (Aug 1960 – Dec 1962)

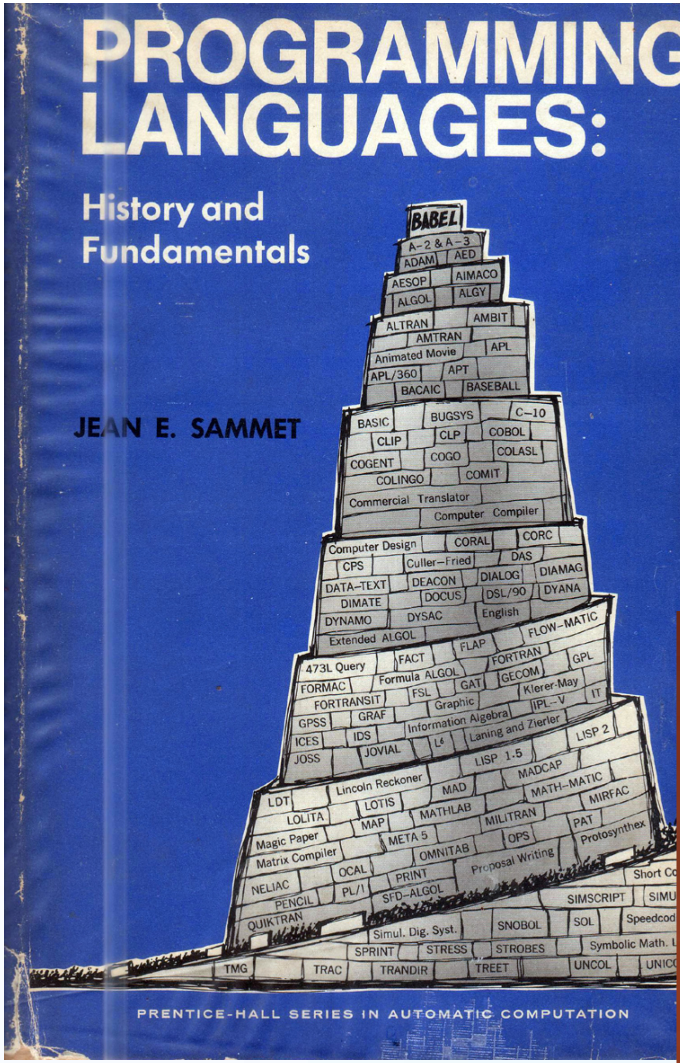
Bill Joy

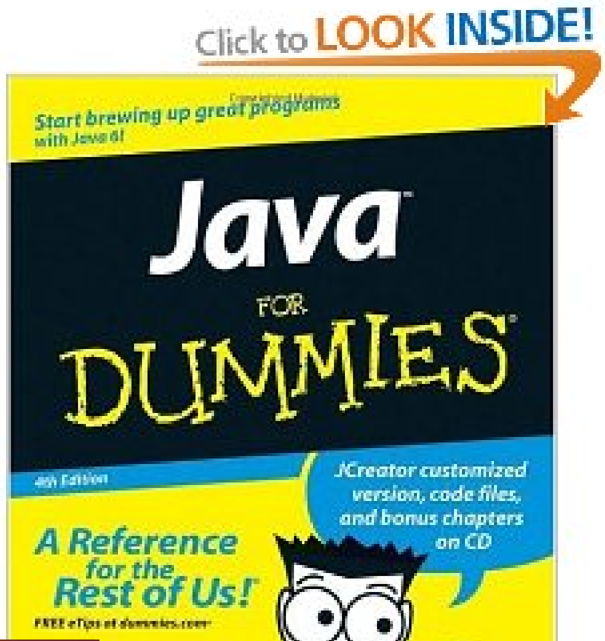


- vi, csh
- Berkeley Unix
  - TCP/IP and VM
- Indirectly
  - Java/Jini/JXTA
  - NFS
  - SPARC

- Time-sharing computer system (fun begins)
- 24-hour opening
- Day and night (1971 Michigan – second year in Berkeley): roughly 10,000 hours (in his words)

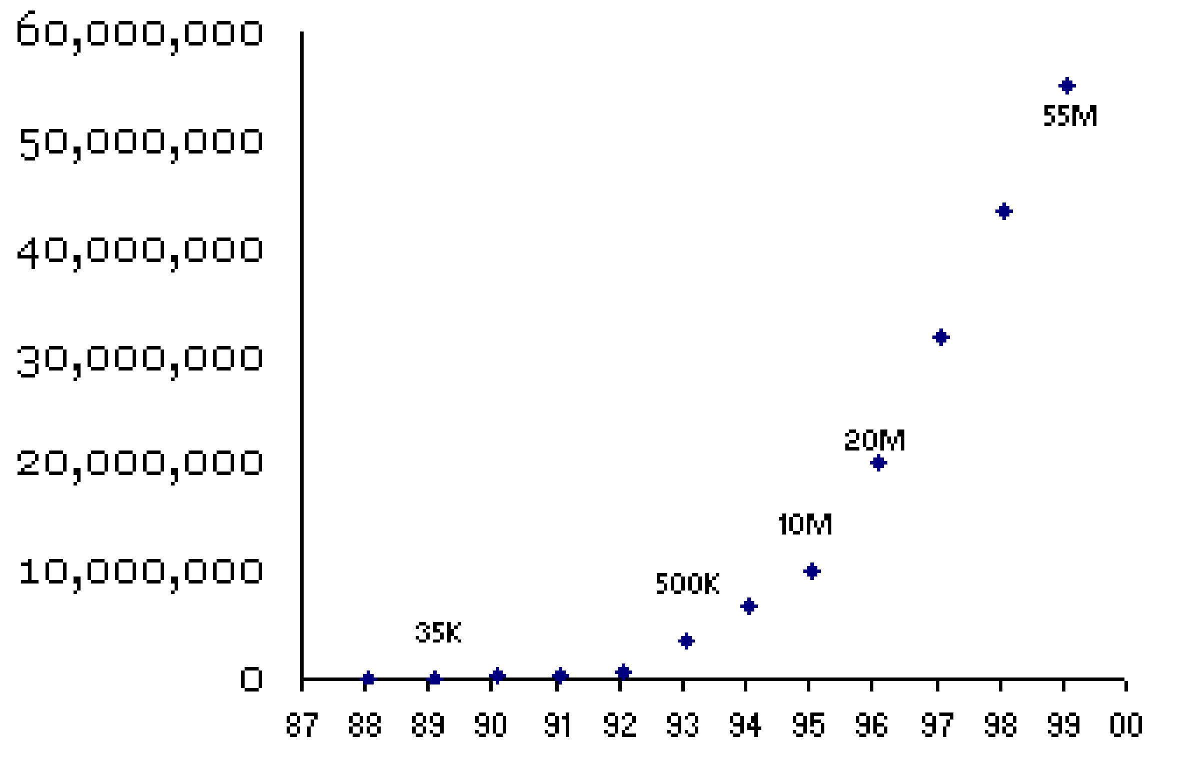
It takes 10 years




Click to LOOK INSIDE!
Start brewing up great programs with Java 6!
Java FOR DUMMIES
4th Edition
JCreator customized version, code files, and bonus chapters on CD
A Reference for the Rest of Us!
FREE eTips at dummies.com

Peter Norvig

It takes 10 years



When Lotus Notes finally shipped, it has been developed for 5 years (1984-1989).

Reactive Arch, Microservices, Containers

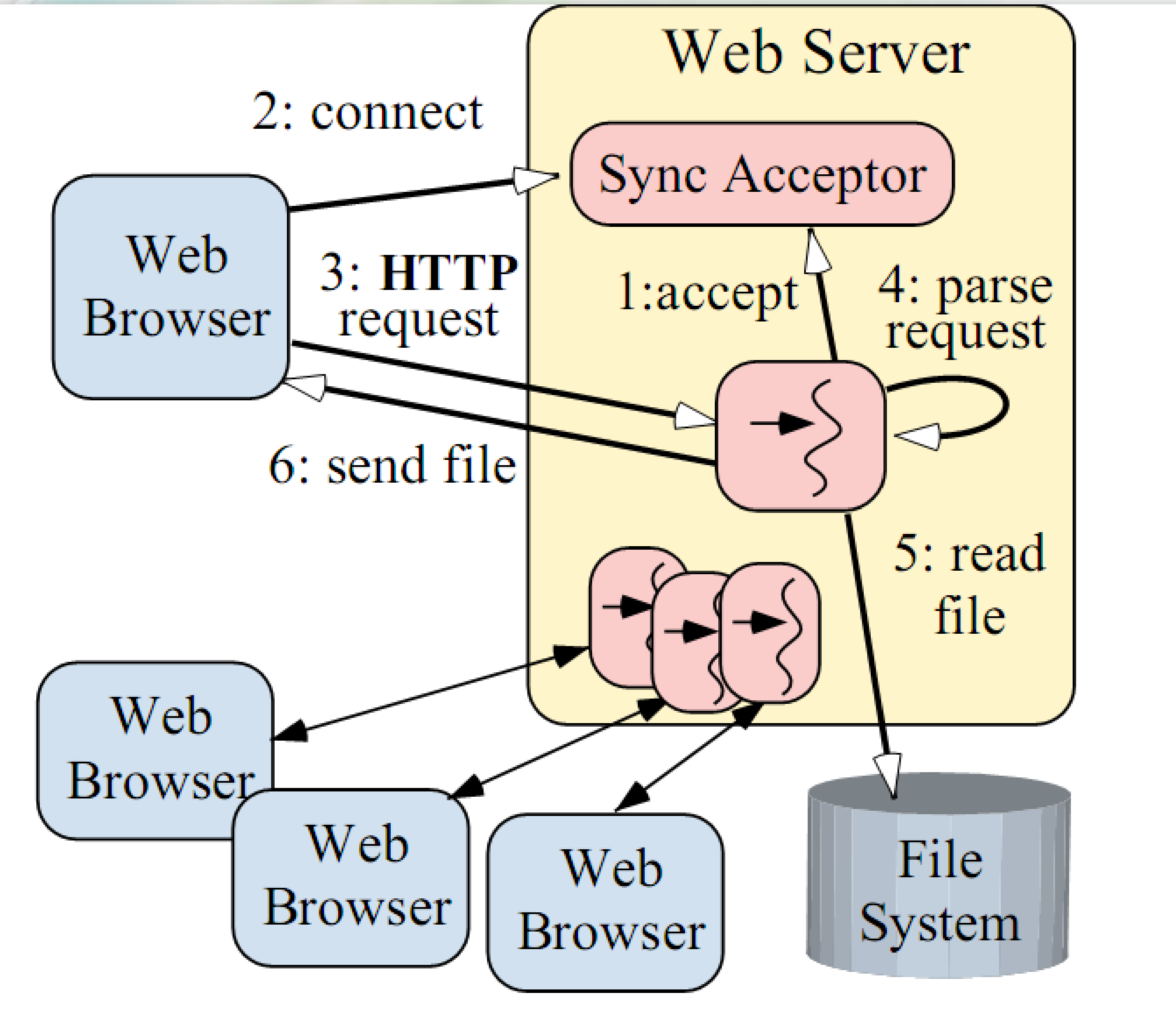
New Endeavors

Thought about Concurrent Models

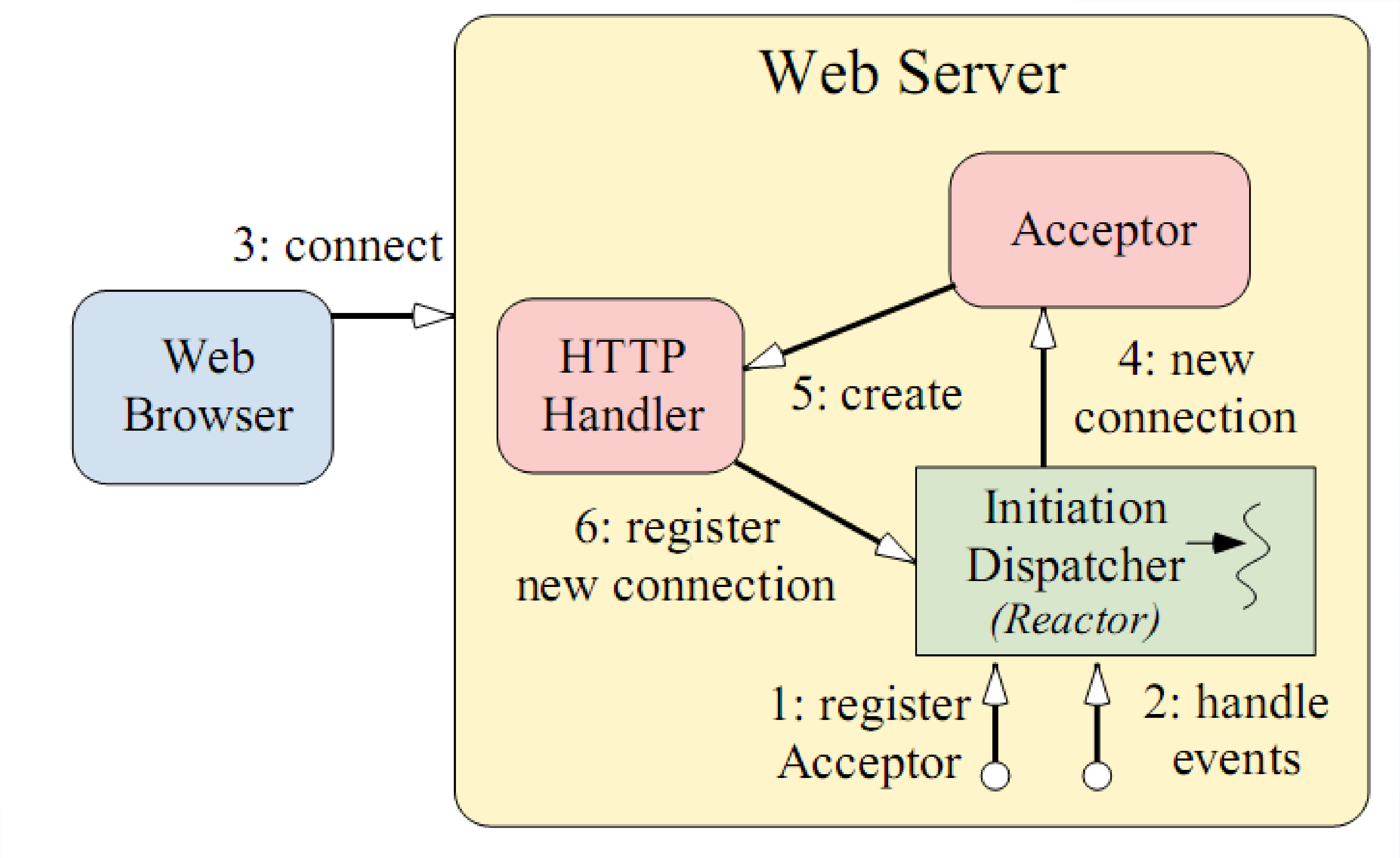
- **Thread**
  - Monitor
  - Scheduling
  - Exported functions
  - Returning from a procedure
  - Blocking procedure call
  - Waiting on condition variables

- **Event-Driven**
  - Event handler
  - Event loop
  - Event types
  - Sending a reply
  - Sending a message
  - Awaiting on messages

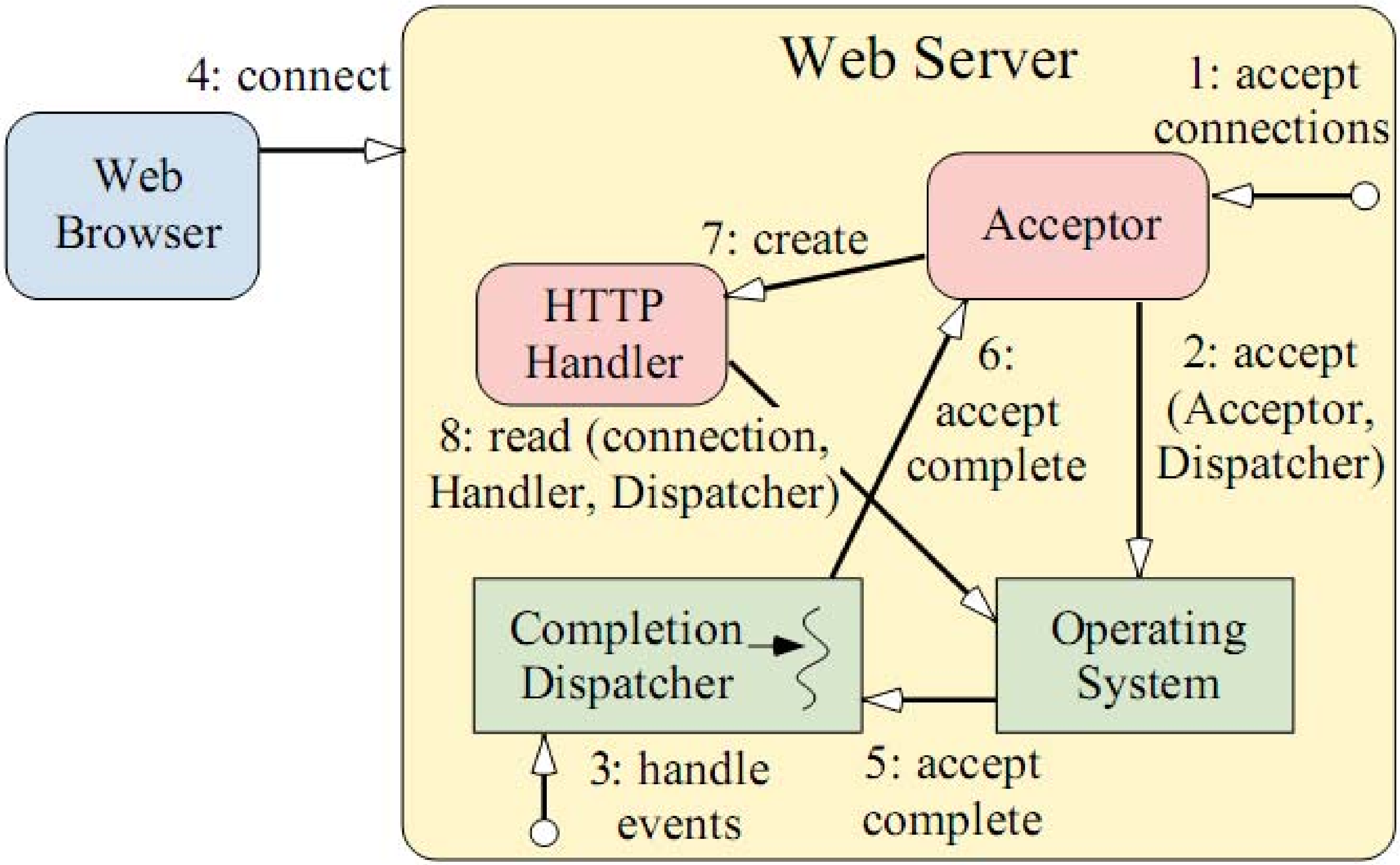
Web App Server (Threads)



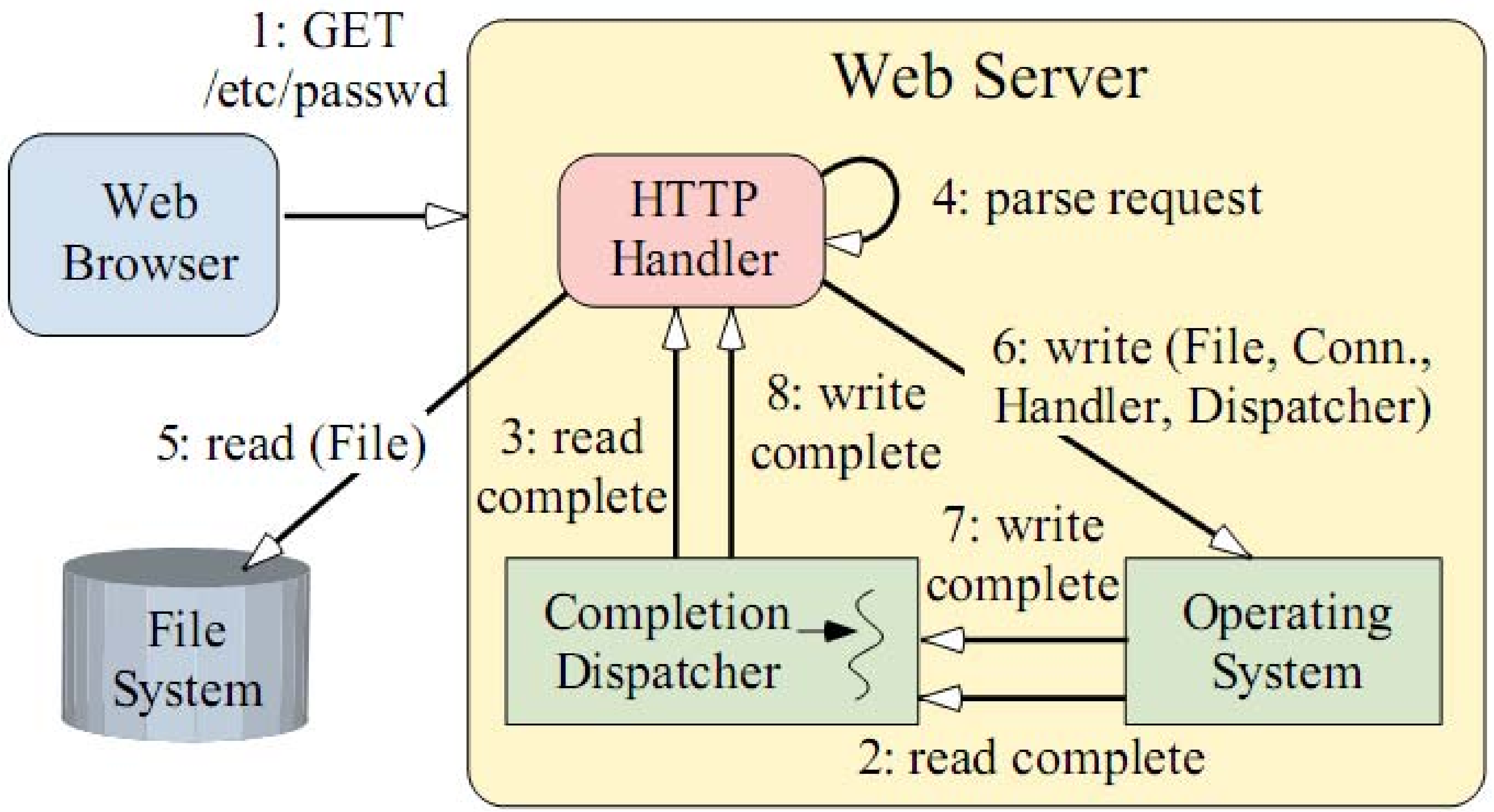
Reactor（Sync, Event-driven)



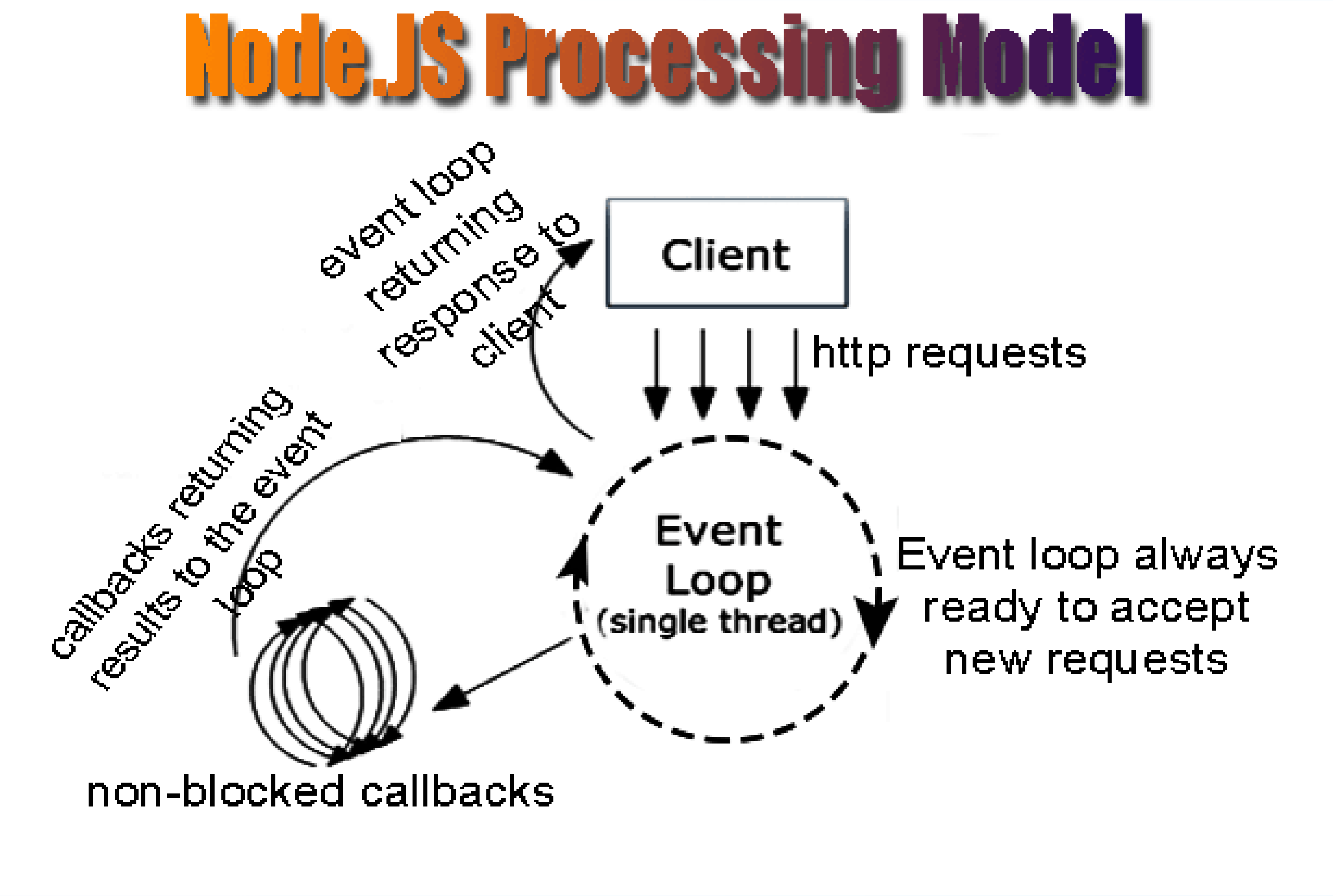
Proactor (client connects)



Proactor (client sends GET)



Event-driven, non-blocking


Node.JS Processing Model
Client
http requests
event loop returning response to client
callbacks returning results to the event loop
Event Loop (single thread)
Event loop always ready to accept new requests
non-blocked callbacks

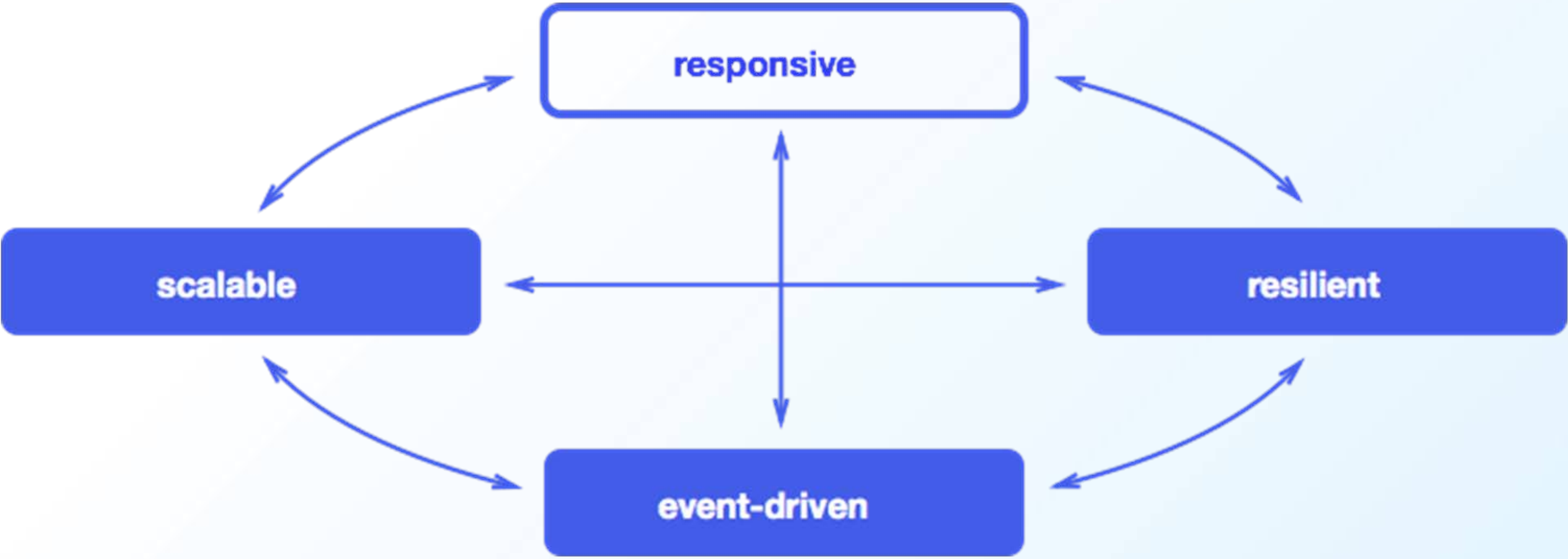
Async Problems

- Memory Leaks
- Race Conditions
- Callback Hell
- Complex State Machines
- Error Handling

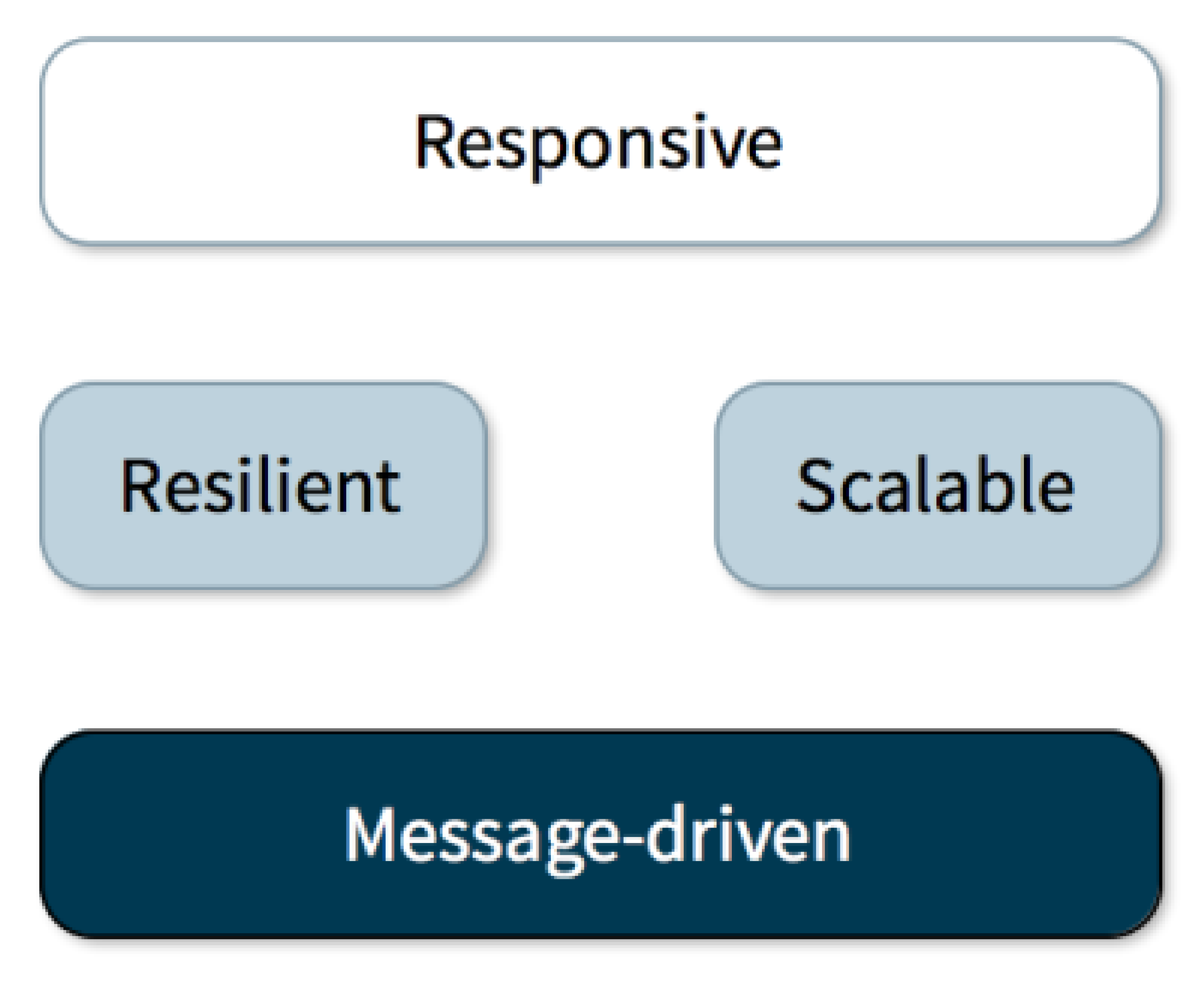
Reactive Architecture

- Architecting with *asynchronous data streams*
  - Everything is a stream
- Functional: Combine, create, filter, map, merge
- Open source projects
  - Google Qbit (library)
  - Spring Reactor
  - Akka (actor framework)

Four Characters



Tiered View



Responsive

Resilient

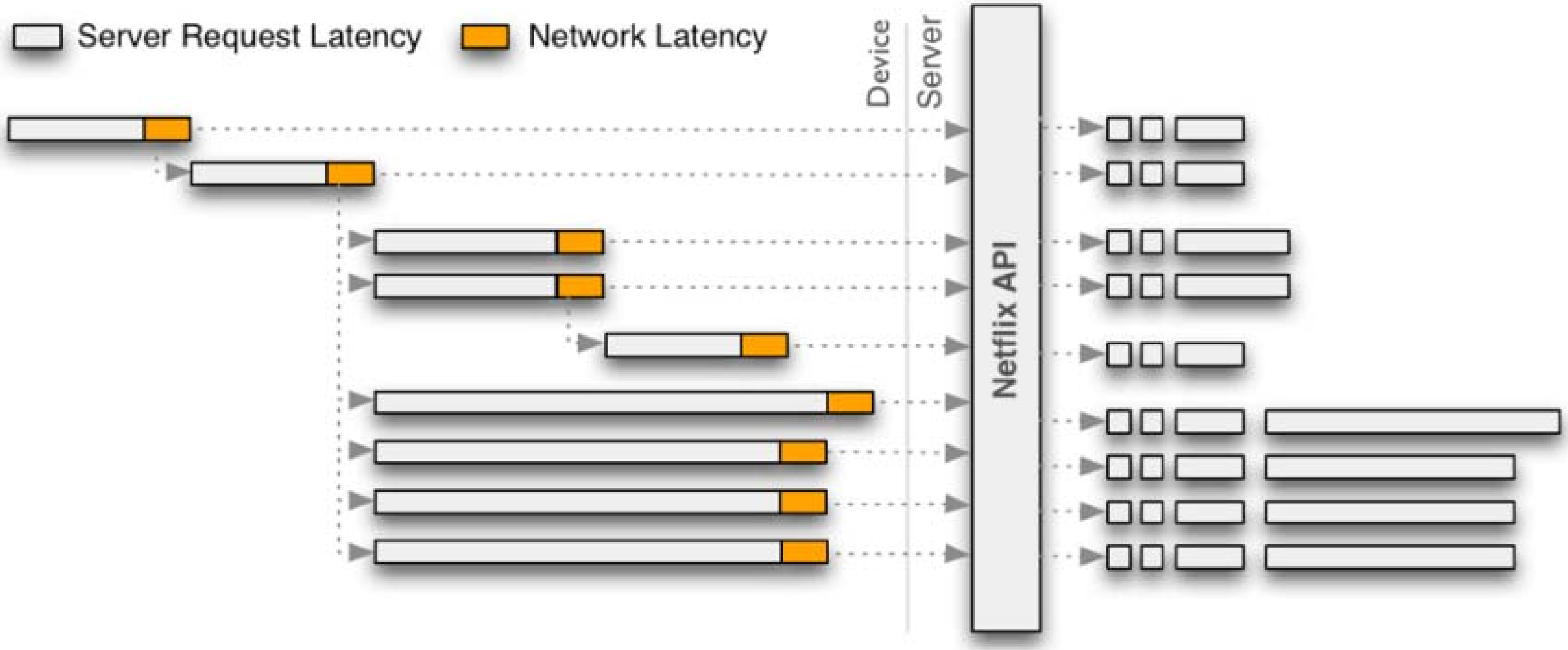
Scalable

Message-driven

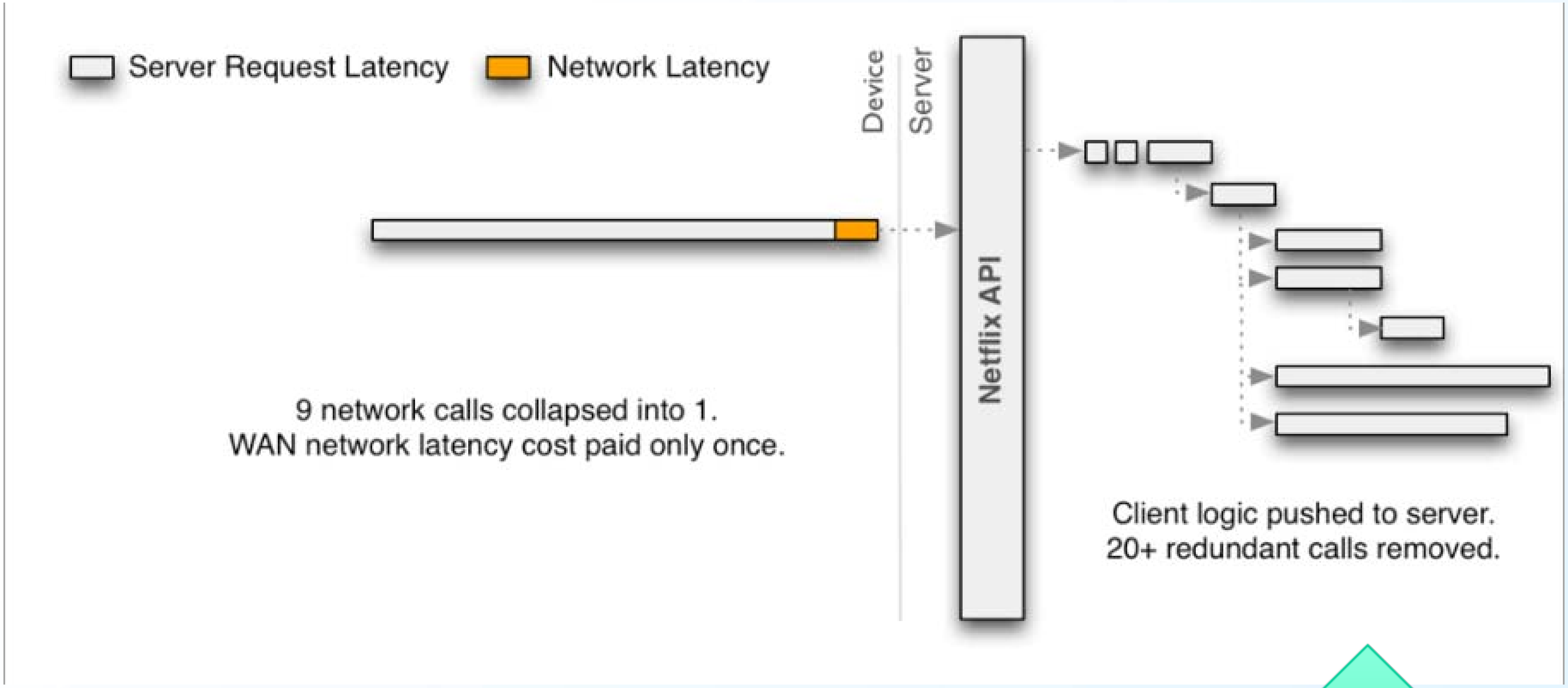
Scalable

- Event-driven
  - GUI
  - Node.js
- Scalable: avoid under-utilization of resources
  - CPU, file handle, db connections, network sockets
  - Thread is a limiting resource
- Resilient: ensure working under threats/faults
  - Messaging: separate error channel
- Responsive: avoid inconsistent user experience
  - Messaging: asynchronous

Netflix Re-arch to be Reactive



Reduce repeated Calls using better/coarser API

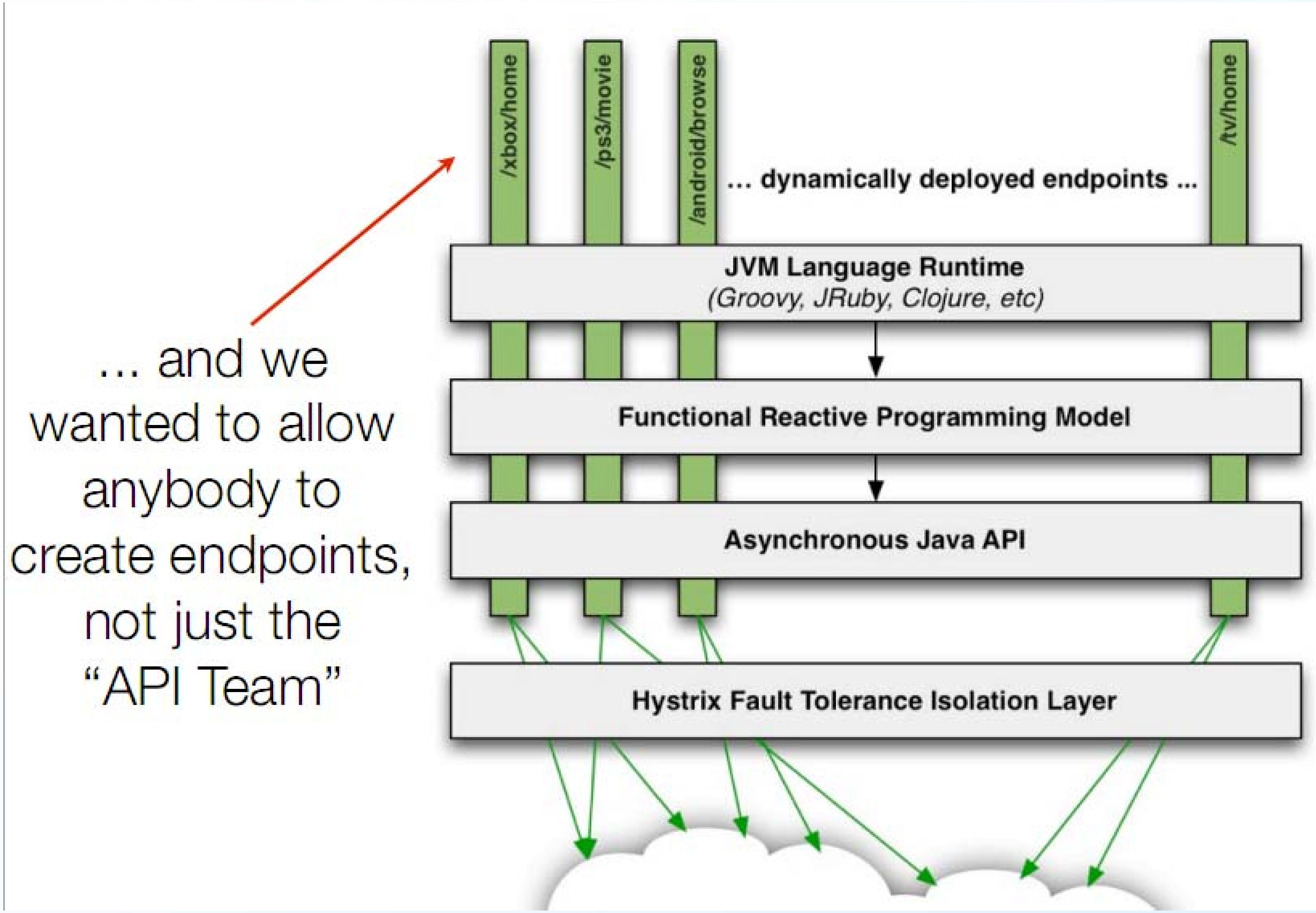


9 network calls collapsed into 1.
WAN network latency cost paid only once.

Client logic pushed to server.
20+ redundant calls removed.

Now use reactive approach for concurrency

Netflix Whole Architecture



… and we wanted to allow anybody to create endpoints, not just the “API Team”

/xbox/home

/ps3/movie

/android/browse

… dynamically deployed endpoints …

/tv/home

JVM Language Runtime
(Groovy, JRuby, Clojure, etc)

Functional Reactive Programming Model

Asynchronous Java API

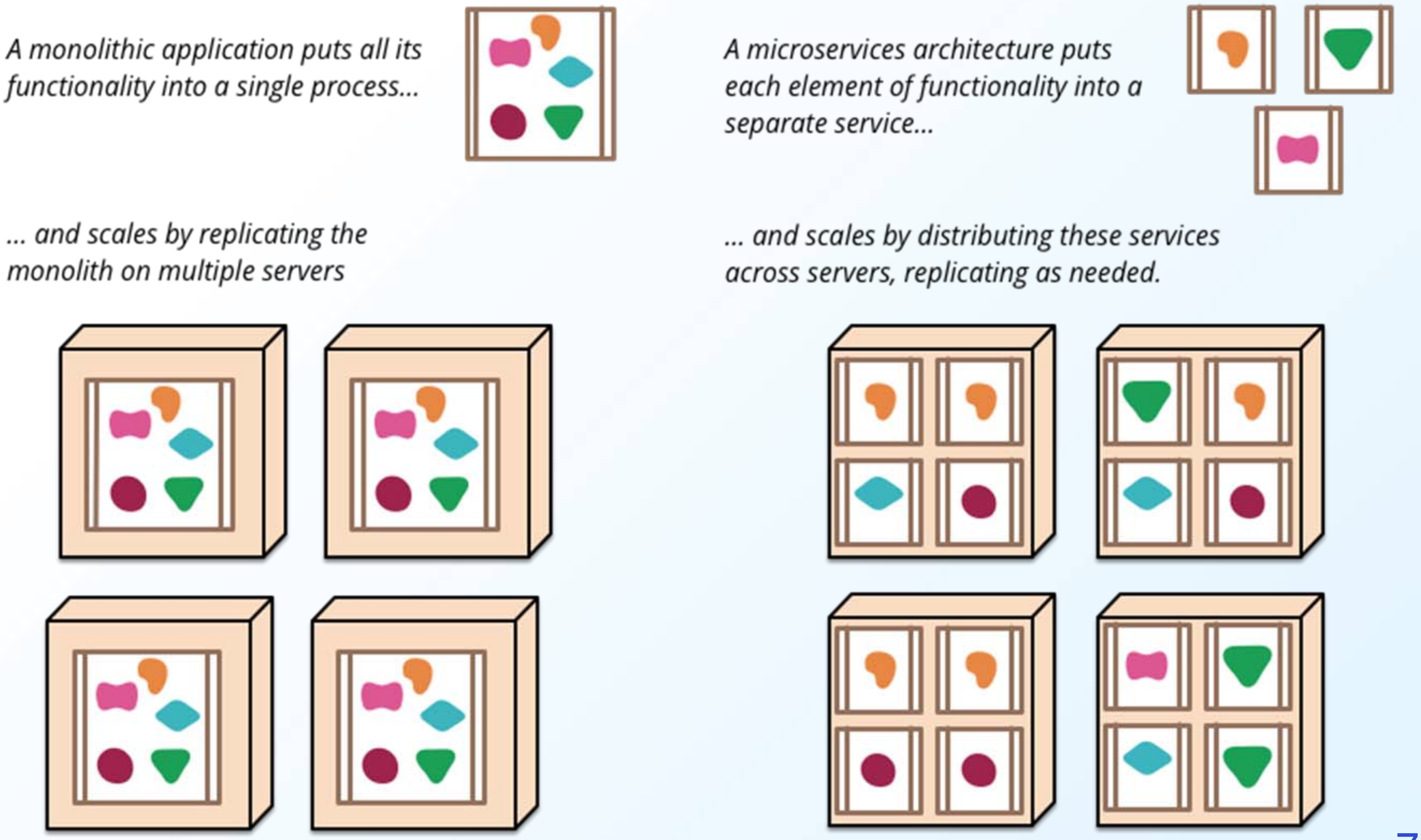
Hystrix Fault Tolerance Isolation Layer

Traditional vs. Reactive

- Iterable
  - Pull
  - T next()
  - throws SomeException
  - return

- Observable
  - Push
  - onNext(T)
  - onError(SomeException)
  - onCompleted()

Microservice Architecture



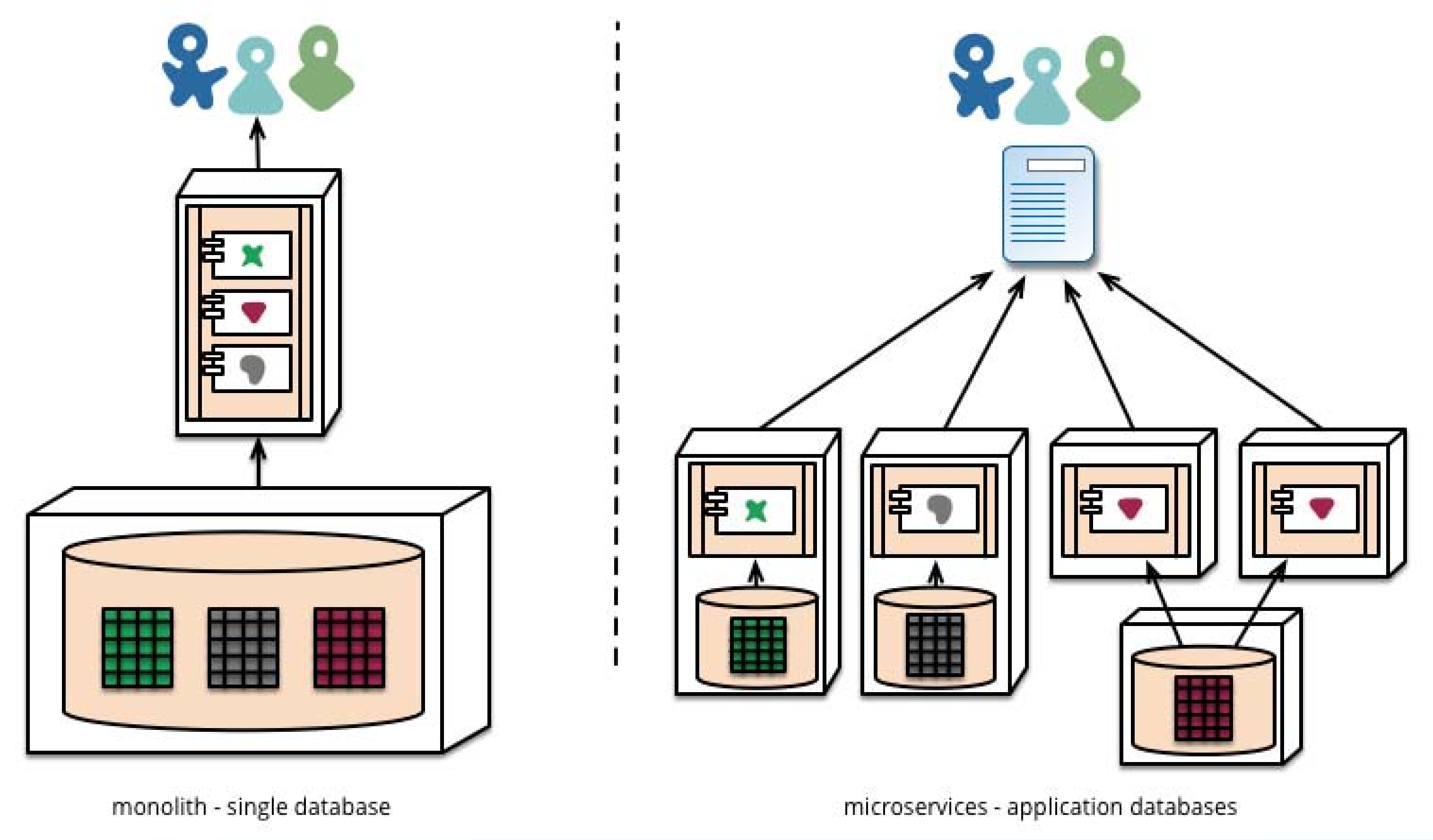
A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers

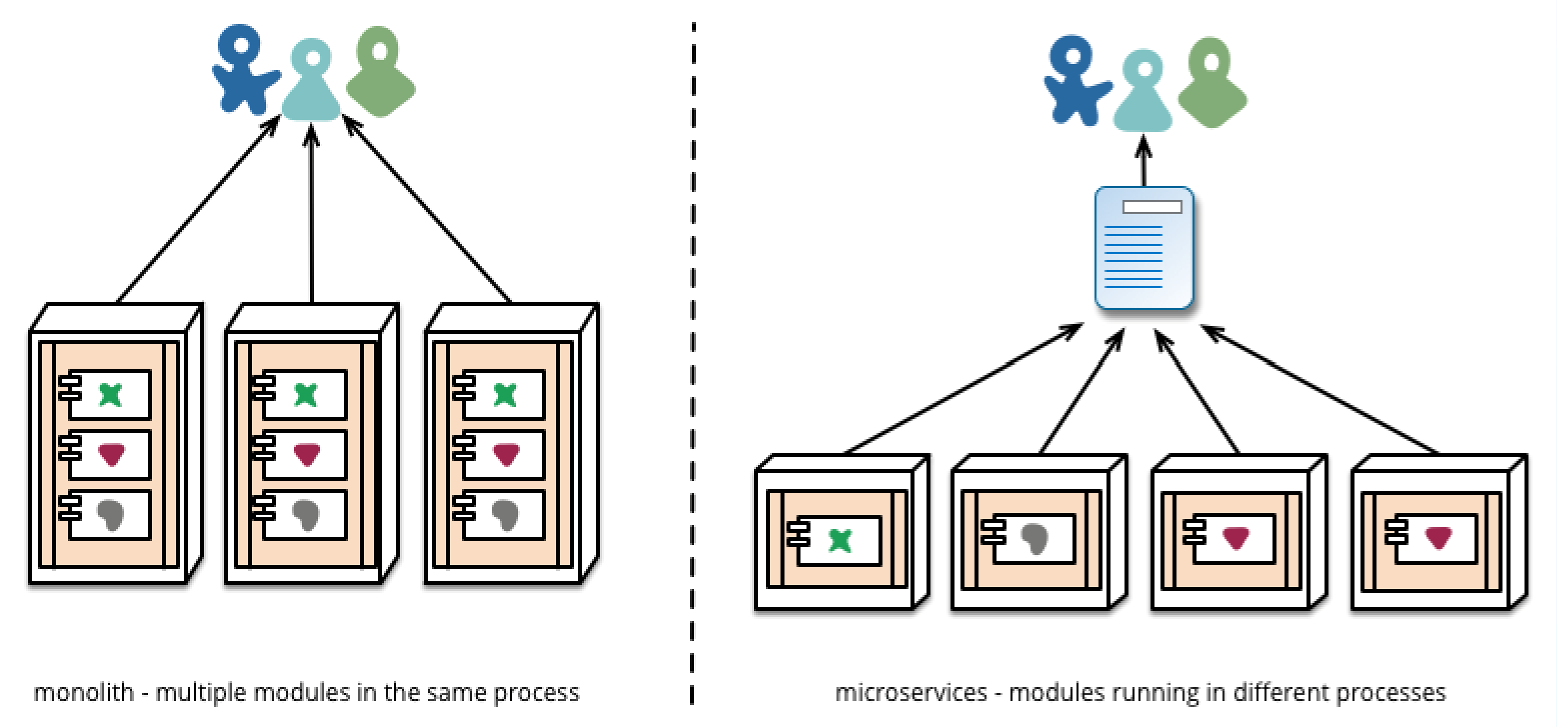
A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.

Monolith vs. Microservice (DB)


monolith - single database
microservices - application databases

Monolith vs. Microservice (process)

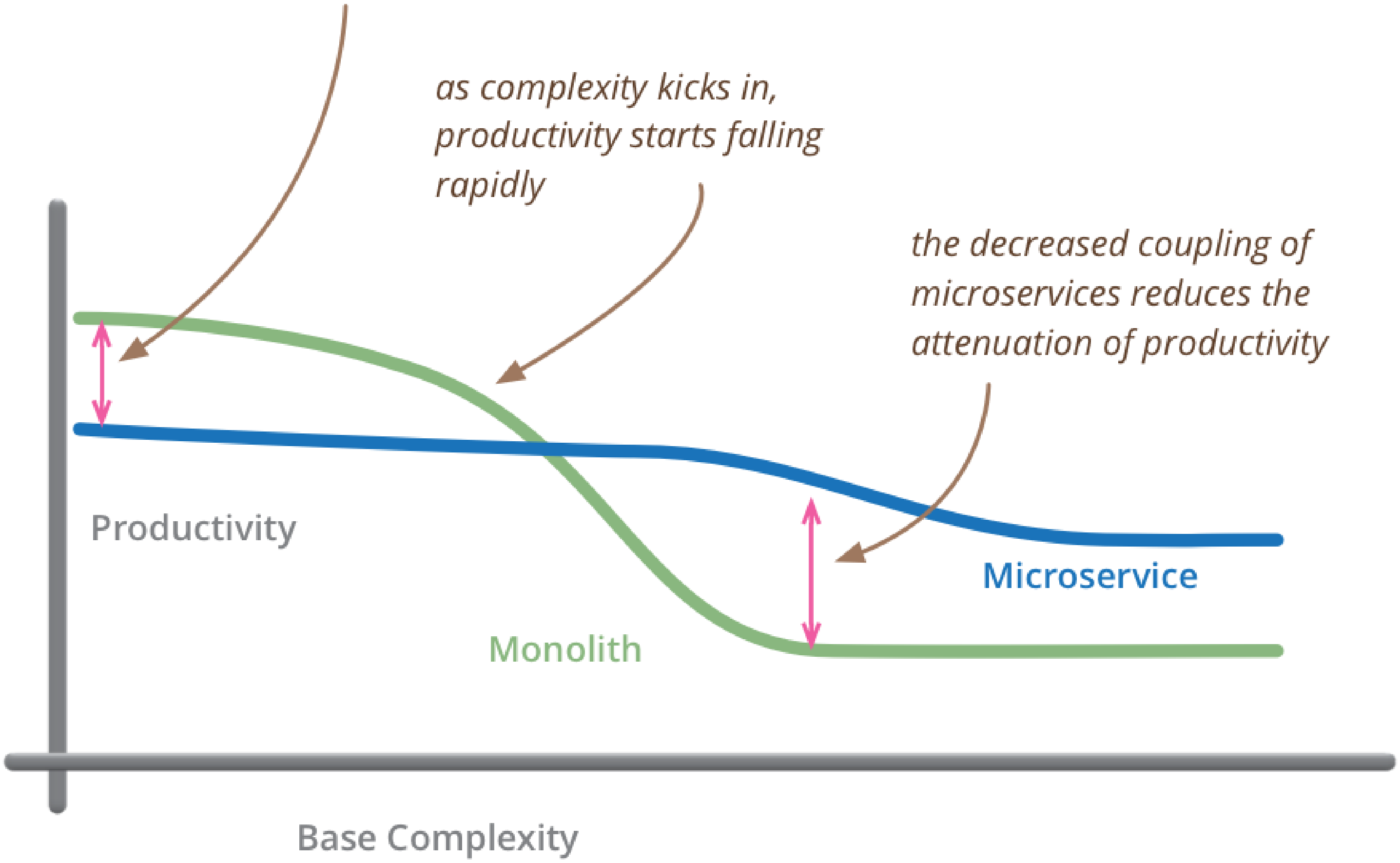

monolith - multiple modules in the same process
microservices - modules running in different processes

Monolith vs. Microservice (productivity)

for less-complex systems, the extra baggage required to manage microservices reduces productivity

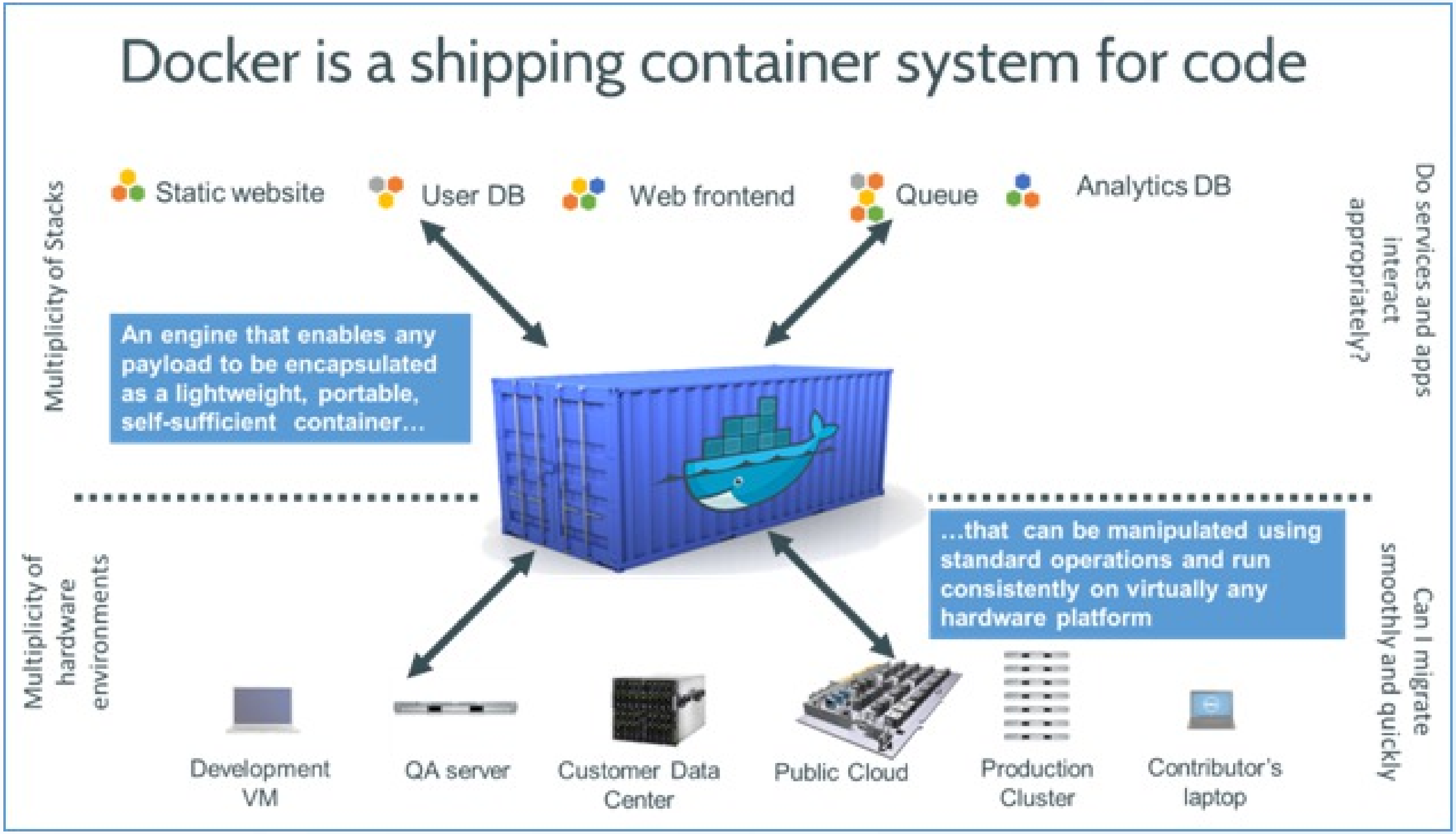
as complexity kicks in, productivity starts falling rapidly

the decreased coupling of microservices reduces the attenuation of productivity



but remember the skill of the team will outweigh any monolith/microservice choice

Docker Container

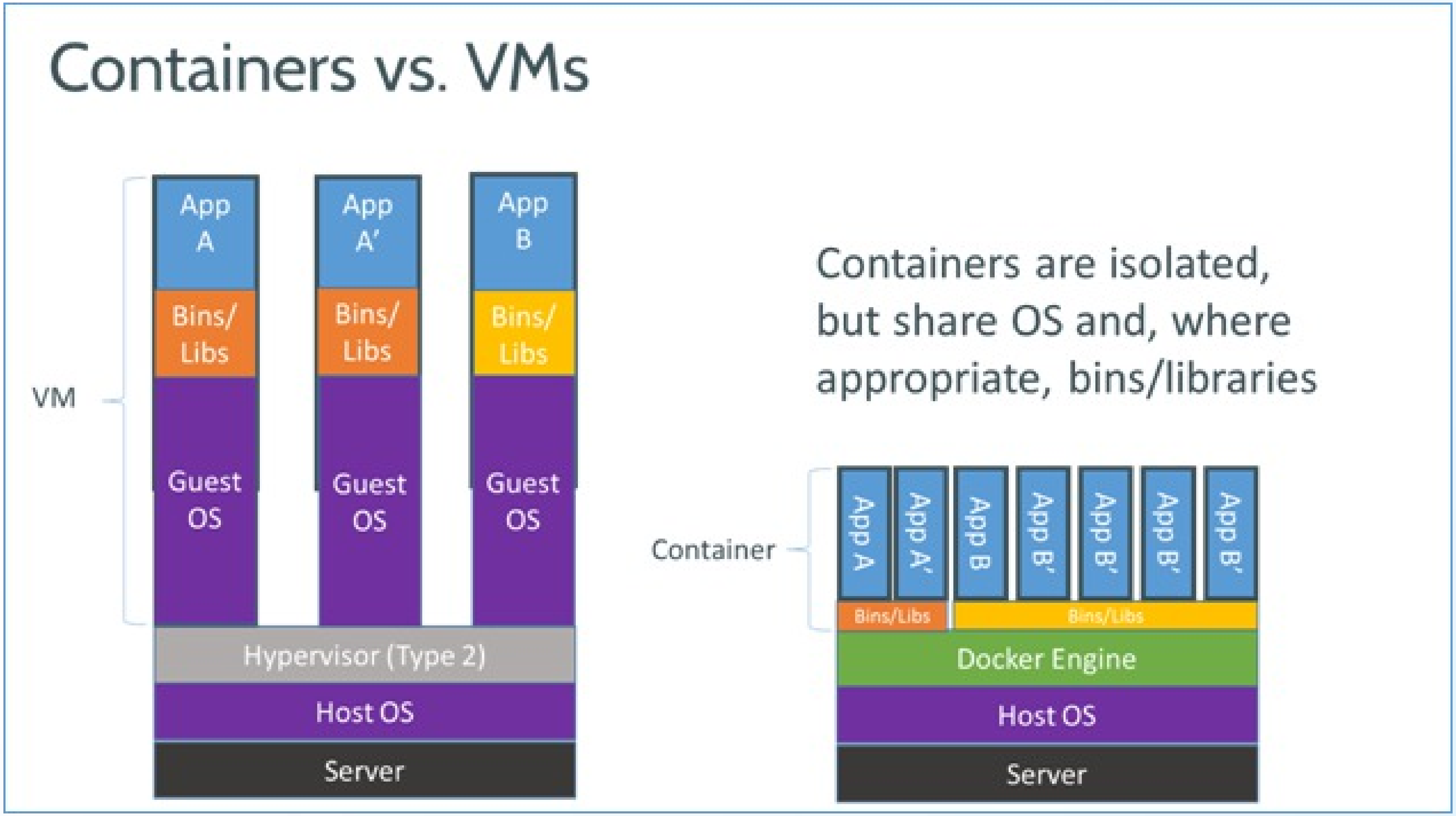


Docker is a shipping container system for code

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container…

…that can be manipulated using standard operations and run consistently on virtually any hardware platform

Much Less Memory and More CPU



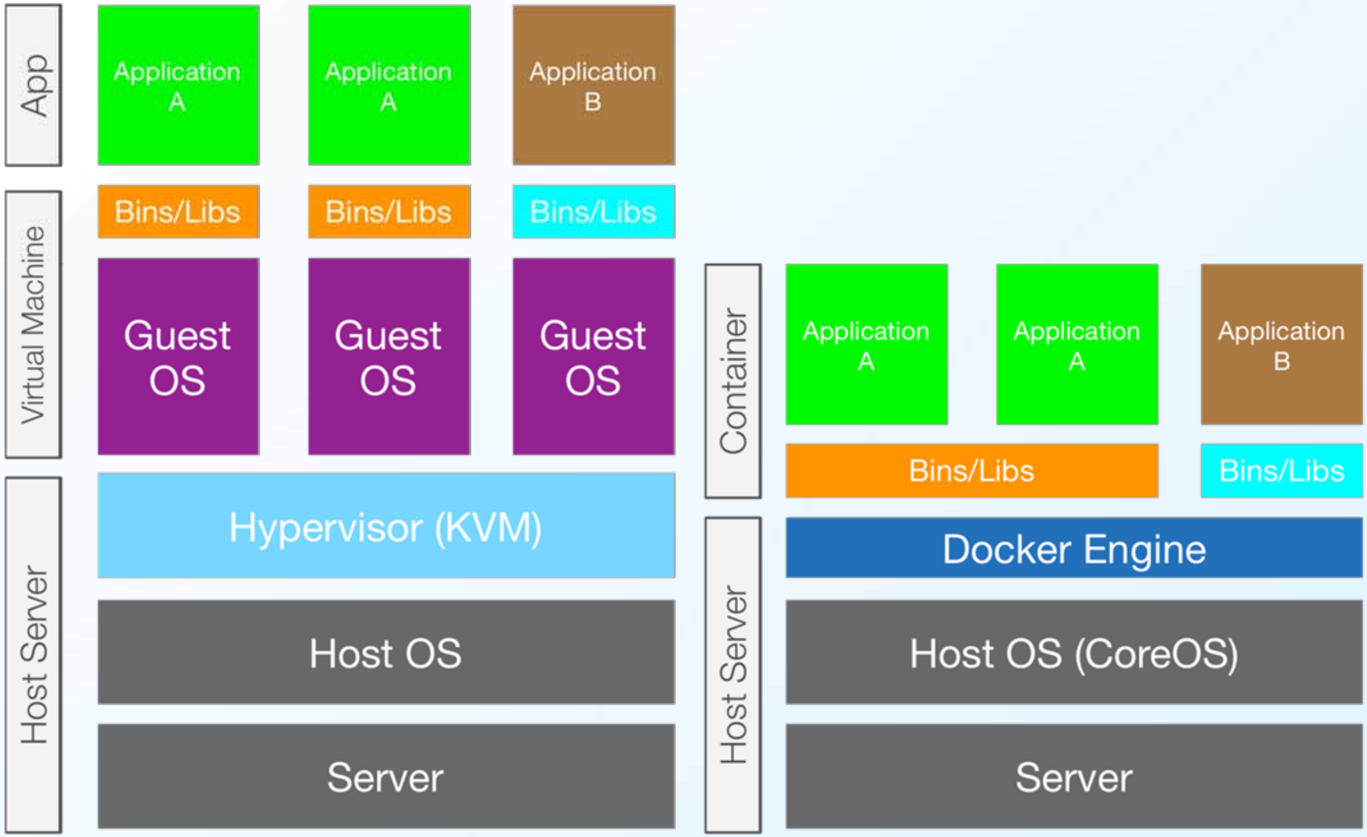
Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

Containers + Micro Services

Better use of resources

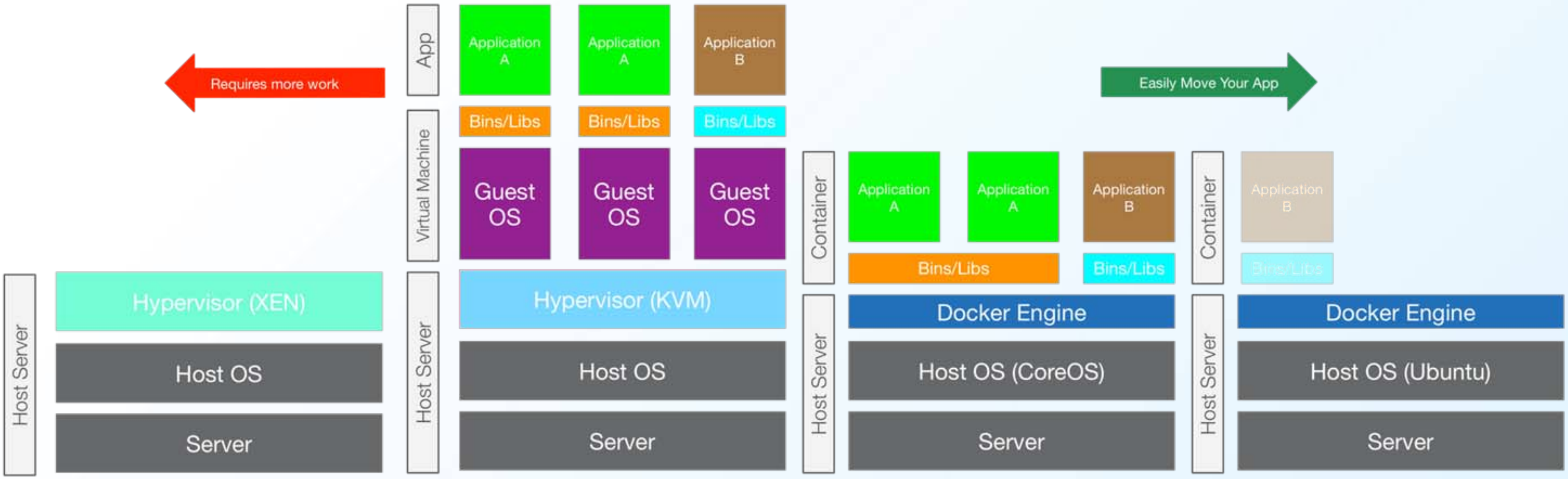
- Containers share the host OS and where appropriate Binaries and Libraries



Containers + Micro Services

Standard container formats such as Docker are cross linux distro compatible

- Incredible easy to move your work load around



And Next Time About Data

Yes, we still like PostgreSQL and hate MongoDB

Q & A