Lancope.
VISION TO SECURE,
INTELLIGENCE TO PROTECT

Micro-services based architecture – a path to “Living Software”?

Jason Chambers

Director of Engineering

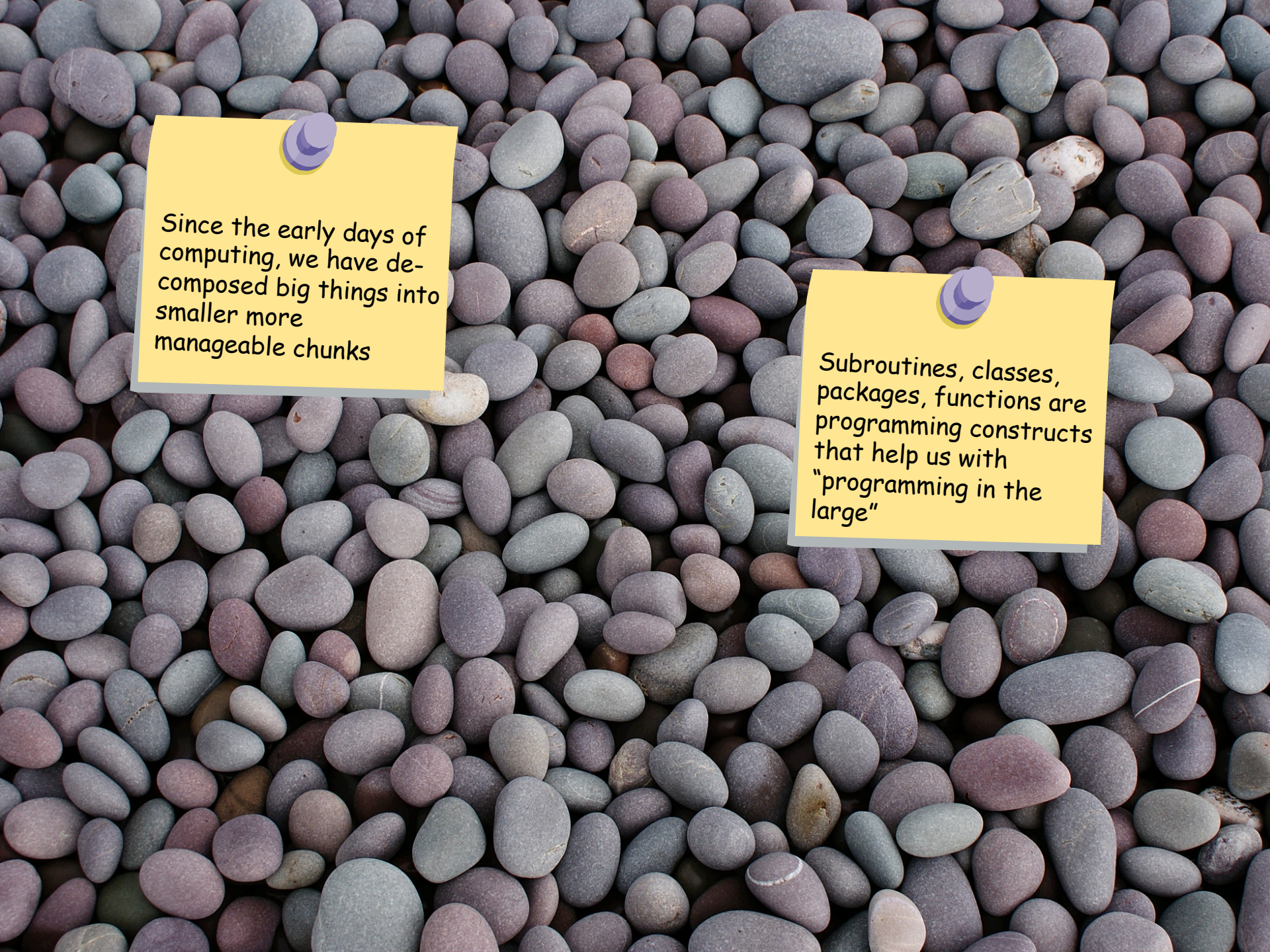
DEVNEXUS

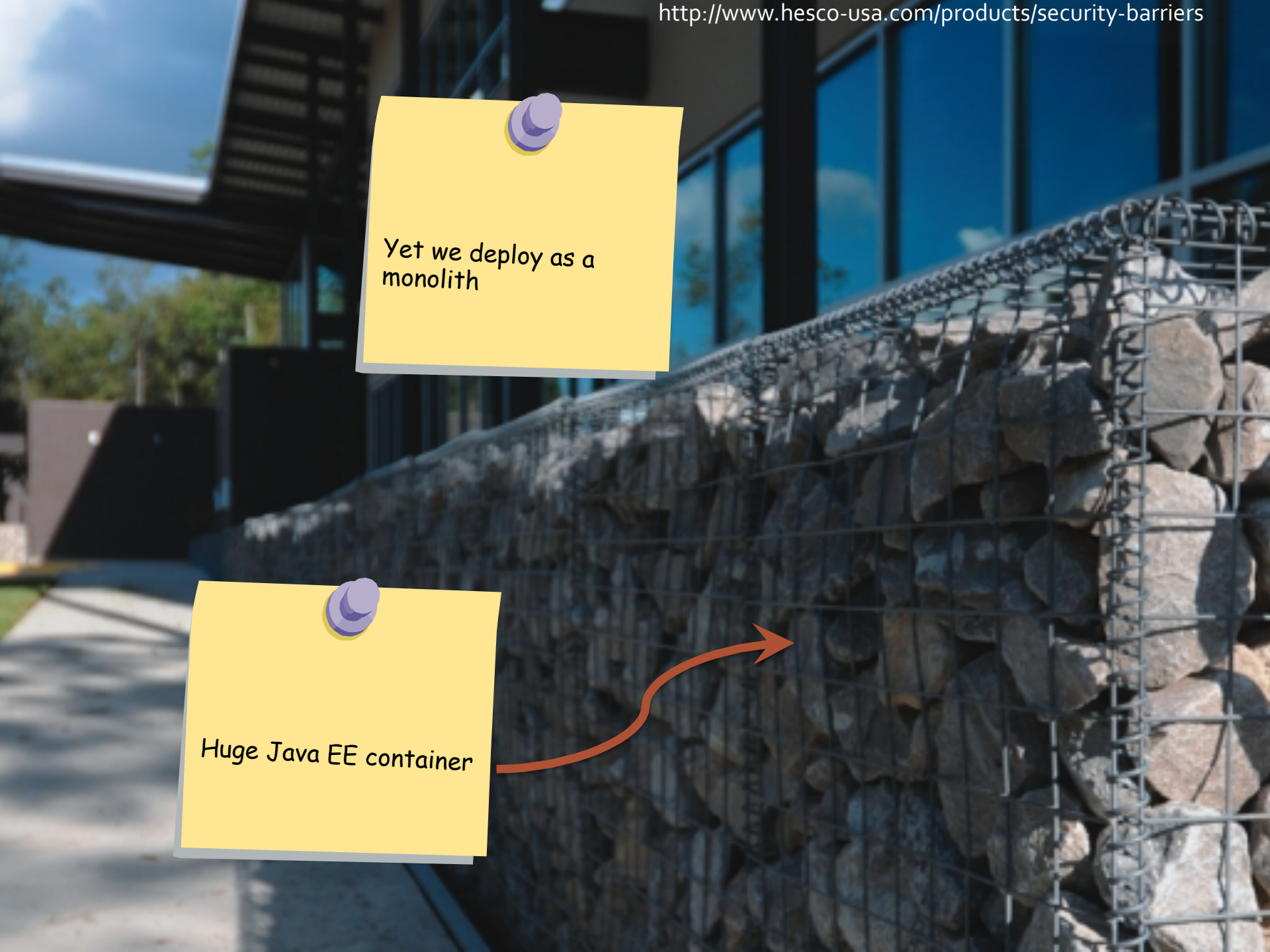
Observations

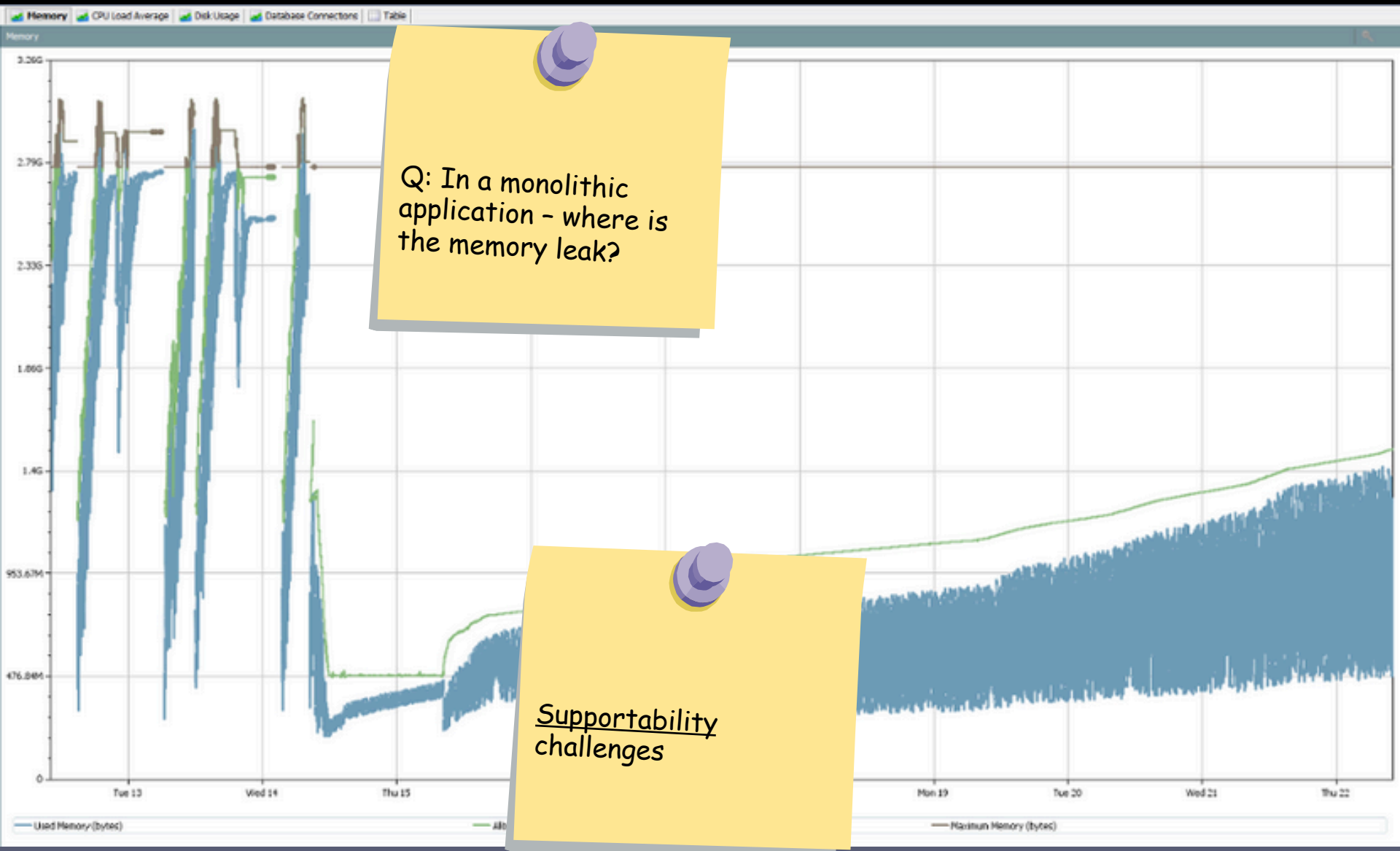
```
10 CLS : PRINT "HAMURABI": PRINT " "
20 PRINT "CREATIVE COMPUTING MORRISTOWN,"
21 PRINT "NEW JERSEY"
30 PRINT " ":PRINT " ": PRINT " ":
80 PRINT "TRY YOUR HAND AT GOVERNING ANCIENT"
81 PRINT "SUMERIA FOR A TEN YEAR TERM OF OFFICE."
90 PRINT " "
95 G=0:R=0
100 Z=0:P=95:S=2800:H=3000:E=H-S
110 Y=3:A=H/Y:I=5:Q=1
210 D=0
215 PRINT " ":PRINT " ": Z=Z+1
216 PRINT "HAMURABI, I BEG TO
217 PRINT "IN THE YEAR ";Z;", ";D
218 PRINT "STARVED, ";I;" CAM
219 P=P+I
225 IF Q>0 THEN GOTO 230
226 P=P/2
227 PRINT "A HORRIBLE PLAG
228 PRINT "HALF THE PEOPL
230 PRINT "POPULATION IS
232 PRINT "THE CITY NOW
235 PRINT "YOU HARVESTE
250 PRINT "THE RATS ATE
260 PRINT "YOU NOW HA
270 IF Z=11 THEN GOTO 8
310 C=1+RND (10):Y=C+17
312 PRINT "LAND IS TRA
313 PRINT "PER ACRE."
320 PRINT "HOW MANY
321 INPUT Q:IF Q<0 TH
322 IF Y*Q <= S THEN
323 GOSUB 710
324 GOTO 320
330 IF Q=0 THEN GOTO 340
331 A=A+Q: S=S-Y*Q: C=0
334 GOTO 400
340 PRINT "HOW MANY ACRES DO YOU WISH TO SELL"
341 INPUT Q: IF Q<0 THEN GOTO 850
342 IF Q<A THEN GOTO 350
343 GOSUB 720
344 GOTO 340
350 A=A-Q: S=S+Y*Q: C=0
400 PRINT " "
410 PRINT "HOW MANY BUSHELS DO YOU WISH TO FEED"
411 PRINT "YOUR PEOPLE": INPUT Q
412 IF Q<0 THEN GOTO 850
418 REM * TRYING TO USE MORE GRAIN THAN IN SILOS
420 IF Q<=S THEN GOTO 430
421 GOSUB 710
422 GOTO 410
430 S=S-Q: C=1: PRINT " "
440 PRINT "HOW MANY ACRES DO YOU WISH TO PLANT"
441 PRINT "WITH SEED": INPUT D
442 IF D=0 THEN GOTO 511
443 IF D<0 THEN GOTO 450
444 REM * PLANTING MORE ACRES THAN YOU OWN
445 IF D<=A THEN GOTO 450
446 GOSUB 720
447 GOTO 440
449 REM * ENOUGH GRAIN FOR SEED
450 IF (D/2)<=S THEN GOTO 455
452 GOSUB 710
453 GOTO 440
454 REM * ENOUGH PEOPLE TO TEND THE CROPS
455 IF D<=10*P THEN GOTO 510
460 PRINT "BUT YOU HAVE ONLY ";P;" PEOPLE TO"
461 PRINT "TO TEND THE FIELDS! NOW THEN,"
470 GOTO 440
510 S=S-D/2
511 GOSUB 800
512 REM * A BOUNTIFUL HARVEST
515 Y=C: H=D*Y: E=0
521 GOSUB 800
522 IF C//2=0 THEN GOTO 530: REM ZERO REMAINDER
523 REM * RATS ARE RUNNING WILD
525 E=S/C
530 S=S-E+H
531 GOSUB 800
532 REM * LET'S HAVE SOME BABIES
533 I=C*(20*A+S)/P/100+1
539 REM * HOW MANY PEOPLE HAD FULL TUMMIES
540 C=Q/20
541 REM * HORRORS, A 15% CHANCE OF PLAGUE
542 Q=7-RND (8): REM Q LESS THAN ZERO IS PLAGUE
550 IF P<C THEN GOTO 210
551 REM * STARVE ENOUGH FOR IMPEACHMENT
552 D=P-C: IF D>P/2 THEN GOTO 560
553 G=((Z-1)*R+D*100/P)Z
555 P=C: G=G+D: GOTO 215
560 PRINT " "
561 PRINT "YOU STARVED ";D;" PEOPLE IN ONE YEAR!"
565 PRINT "DUE TO THIS MISMANAGE
566 PRINT "NOT ONLY BEEN IMPEAC
567 PRINT "OUT OF THE OFFICE BUT
568 PRINT "ALSO BEEN DECRALED N
569 GOTO 990
710 PRINT "HAMURABI, THINK AGAI
711 PRINT S;" BUSHELS OF GRAIN. N
712 RETURN
720 PRINT "HAMURABI, THINK AGAI
721 PRINT A;" ACRES. NOW THEN,"
730 RETURN
800 C=1+RND (5)
801 RETURN
850 PRINT " ": PRINT " "
851 PRINT "HAMURABI, I CANNOT
855 PRINT "GET YOURSELF ANOTH
857 GOTO 990
860 PRINT "IN YOUR 10-YEAR TERM OF OFFICE,"
861 PRINT R;" PERCENT OF THE POPULATION STARVED"
862 PRINT "PER YEAR ON AVERAGE, I.E. A TOTAL OF"
865 PRINT G;" PEOPLE DIED!": L=A/P
870 PRINT "YOU STARTED WITH 10 ACRES PER PERSON"
871 PRINT "AND ENDED WITH ";L;" ACRES PER"
875 PRINT "PERSON.": PRINT " "
880 IF R>10 THEN GOTO 940
885 IF L<9 THEN GOTO 940
895 IF R>3 THEN GOTO 960
896 IF L<10 THEN GOTO 960
900 PRINT "A FANTASTIC PERFORMANCE! CHARLEMAGNE"
901 PRINT "DISREALI, AND JEFFERSON COMBINED"
902 PRINT "COULD NOT HAVE DONE BETTER!"
905 GOTO 990
940 PRINT "YOUR HEAVY-HANDED PERFORMANCE SMACKS"
941 PRINT "OF NERO AND IVAN IV. THE PEOPLE"
942 PRINT "(REMAINING) FIND YOU AN
UNPLEASANT"
943 PRINT "RULER, AND, FRANKLY, HATE YOUR
GUTS!"
950 GOTO 990
960 PRINT "YOUR PERFORMANCE COULD HAVE
BEEN"
961 PRINT "SOMEWHAT BETTER, BUT REALLY
WASN'T"
962 PRINT "TOO BAD AT ALL. "
963 PRINT 1+RND (P);" PEOPLE WOULD DEARLY
LIKE"
964 PRINT "TO SEE YOU ASSASINATED BUT WE ALL"
965 PRINT "HAVE OUR TRIVIAL PROBLEMS."
990 PRINT " "
                                    FOR NOW.": PRINT " "
                                    G***": REM * DEBUG
                                    OPLE DIED ";G
                                    OPULATION STARVED ";R
                                    EAR IS ";Z
                                    ON IS ";P
                                    ORAGE ";S
                                    AS ";H
                                    LD, PRICE PER ACRE ";Y
1100 PRINT "PEOPLE CAME TO CITY ";I
1110 PRINT "CURRENT INPUT OR RND VALUE ";Q
1120 PRINT "PEOPLE STARVED OR ACRES PLANTED
";D
1130 PRINT "RND VALUE FROM 1 TO 10 ";C
1140 PRINT "ACRES PER PERSON ";L
1150 PRINT "RND VALUE FROM 1 TO 5 OR FULL"
1151 PRINT "TUMMIES ";C
1160 PRINT "***DEBUG***": REM END OF DEBUG
1170 RETURN
```

This is similar in structure to the first BASIC programs I wrote as a kid

I still come across shell-scripts written like this <sigh/>

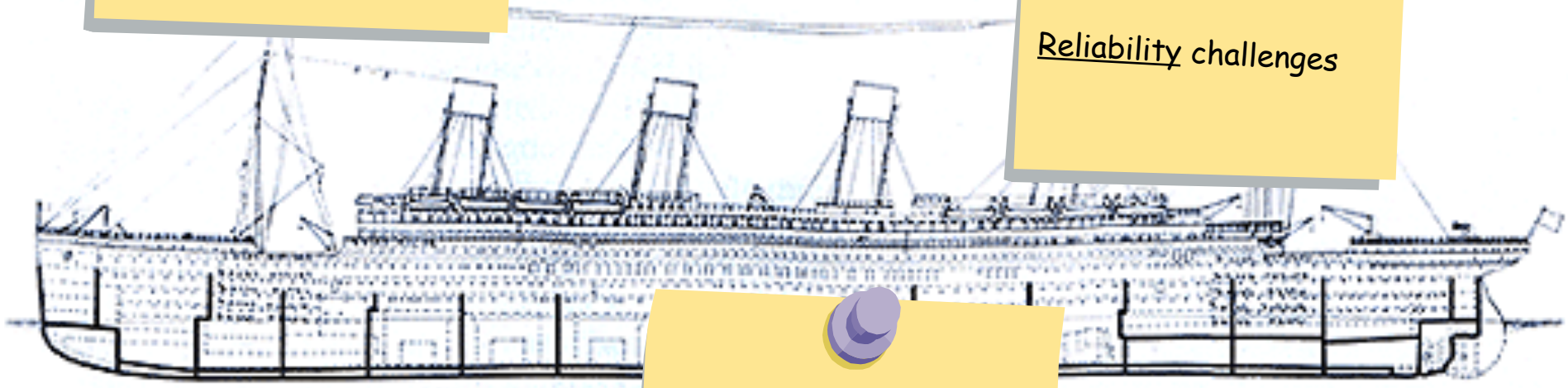

Since the early days of computing, we have decomposed big things into smaller more manageable chunks
Subroutines, classes, packages, functions are programming constructs that help us with "programming in the large"


http://www.hesco-usa.com/products/security-barriers
Yet we deploy as a monolith
Huge Java EE container

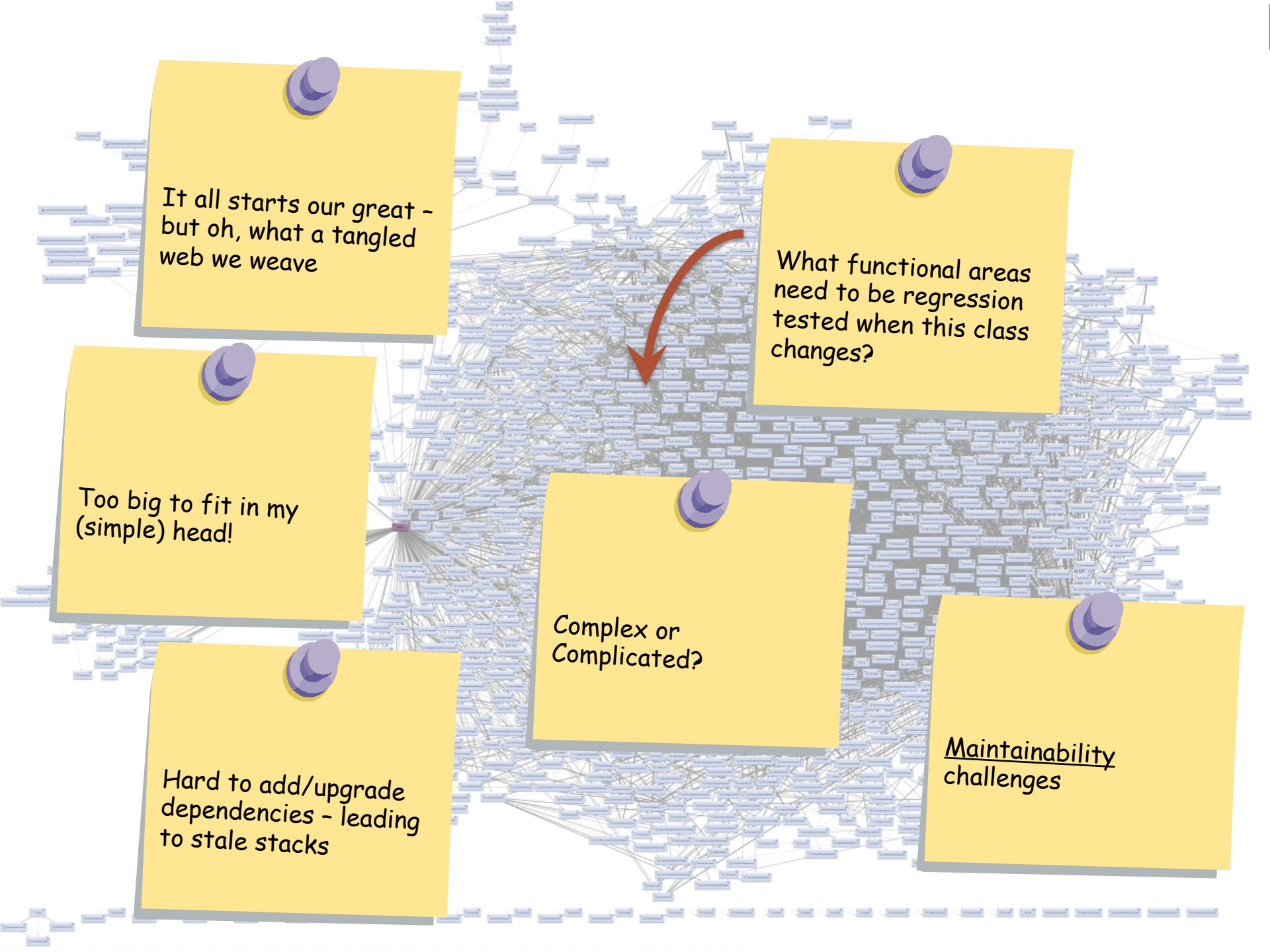

Memory
CPU Load Average
Disk Usage
Database Connections
Table
Q: In a monolithic application – where is the memory leak?
Supportability challenges
Used Memory (bytes)
Maximum Memory (bytes)

Q: In a monolithic application – what fails when you see OutOfMemoryException?

Reliability challenges



Do you see isolated or catastrophic cascading failure?


It all starts our great - but oh, what a tangled web we weave
What functional areas need to be regression tested when this class changes?
Too big to fit in my (simple) head!
Complex or Complicated?
Maintainability challenges
Hard to add/upgrade dependencies - leading to stale stacks

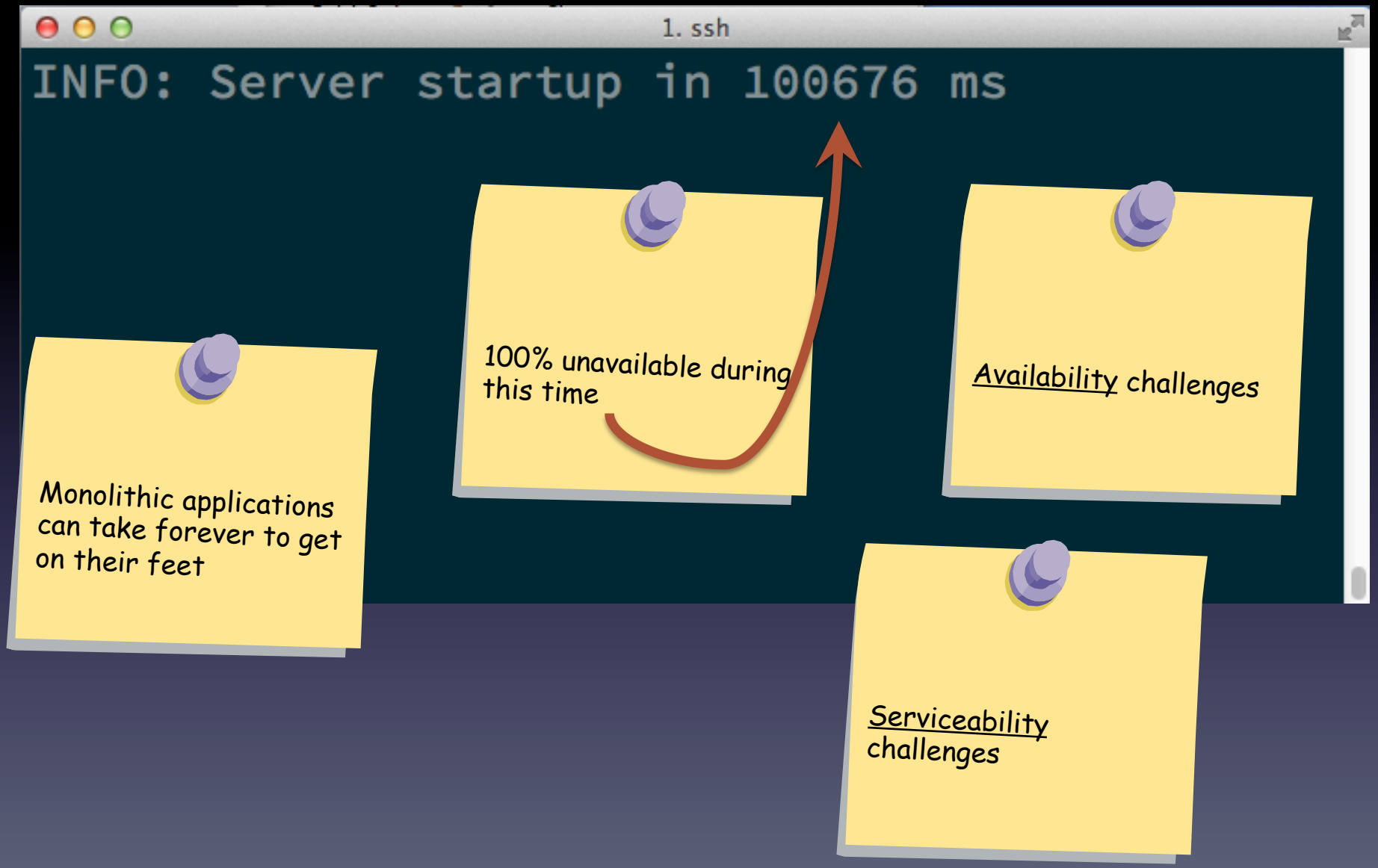
Do monolithic architectures impede Continuous Delivery?

"Adopting Continuous Delivery: Adjusting your Architecture",
Rachel Laycock, Qcon,
http://www.infoq.com/presentations/adopting-continuous-delivery

```
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 19:14.974s
[INFO] Finished at: Wed Feb 05 12:11:44 EST 2014
[INFO] Final Memory: 781/187M
[INFO] ------------------------------------------------------------------------
```

Monolithic applications can take forever to build

Productivity killer


1. ssh
INFO: Server startup in 100676 ms
100% unavailable during this time
Availability challenges
Monolithic applications can take forever to get on their feet
Serviceability challenges

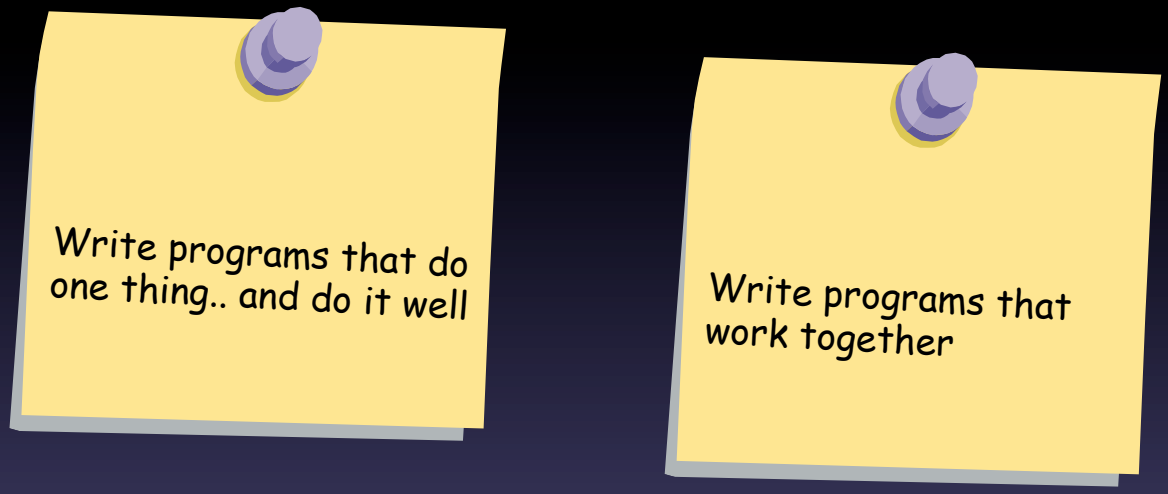

Too many cooks spoil the broth
Impedes ability to scale the team

The challenges of monolithic applications


CITY OF
Atlanta
Department of Watershed Management
Inside
Newsroom
Customer Service
Home
Documents and Reports
Offices
History
Clean Water Atlanta
Better Building Challenge
Floodplain Maps
Home » Inside DWM » History
History
1
DWM was formed in September 2002 to manage the City of Atlanta's
utility operations: drinking water, wastewater and stormwater systems. As part
of this operation, DWM manages one of the largest water capital improvement
programs in the country at an estimated cost of approximately $4 billion. In
addition, the wastewater component of the capital program is largely controlled
by two Federal Consent Decrees that have perhaps the most stringent and
demanding schedule and performance requirements in the country. These
Consent Decrees were ordered by the Federal Court in 1998 and 1999 to
compel the City of Atlanta to address the conditions of the wastewater syste
that had been significantly underfunded and seriously under-maintained for
decades. Watershed Management completed all construction for the first
Consent Decree in 2008 in what was termed by the Federal judge a
This is just infrastructure stuff – why should we care?

Can micro-services help to address the challenges of monolithic applications?

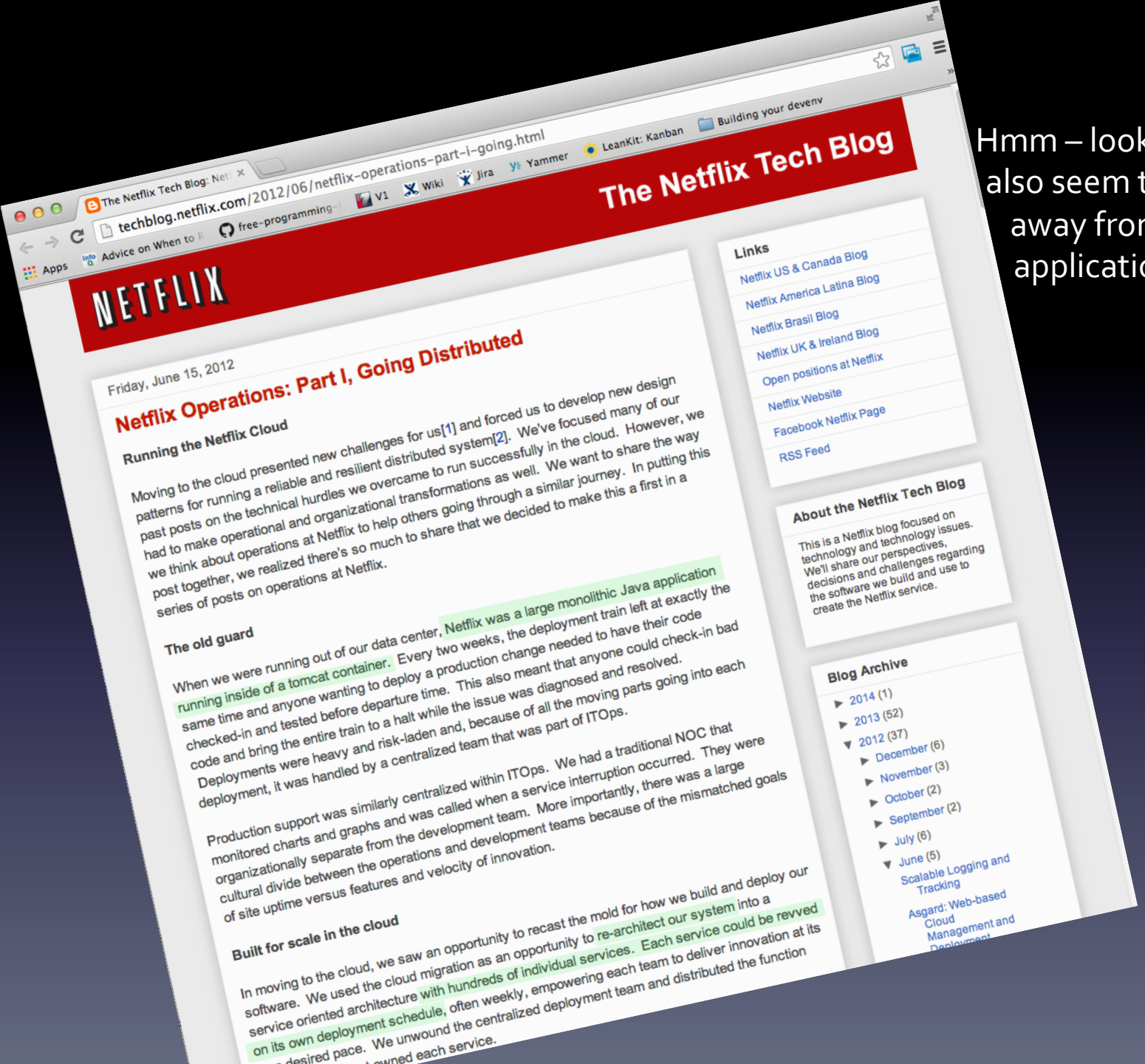
The UNIX philosophy




```
#!/bin/sh
cat test.txt | tr ' ' '\n' | sort | uniq -c | sort -rn
```

"Micro Services: Java, the Unix Way", QCon San Francisco 2012, James Lewis

The Micro-services architecture is an approach – the choice of platform and programming language is merely an implementation detail

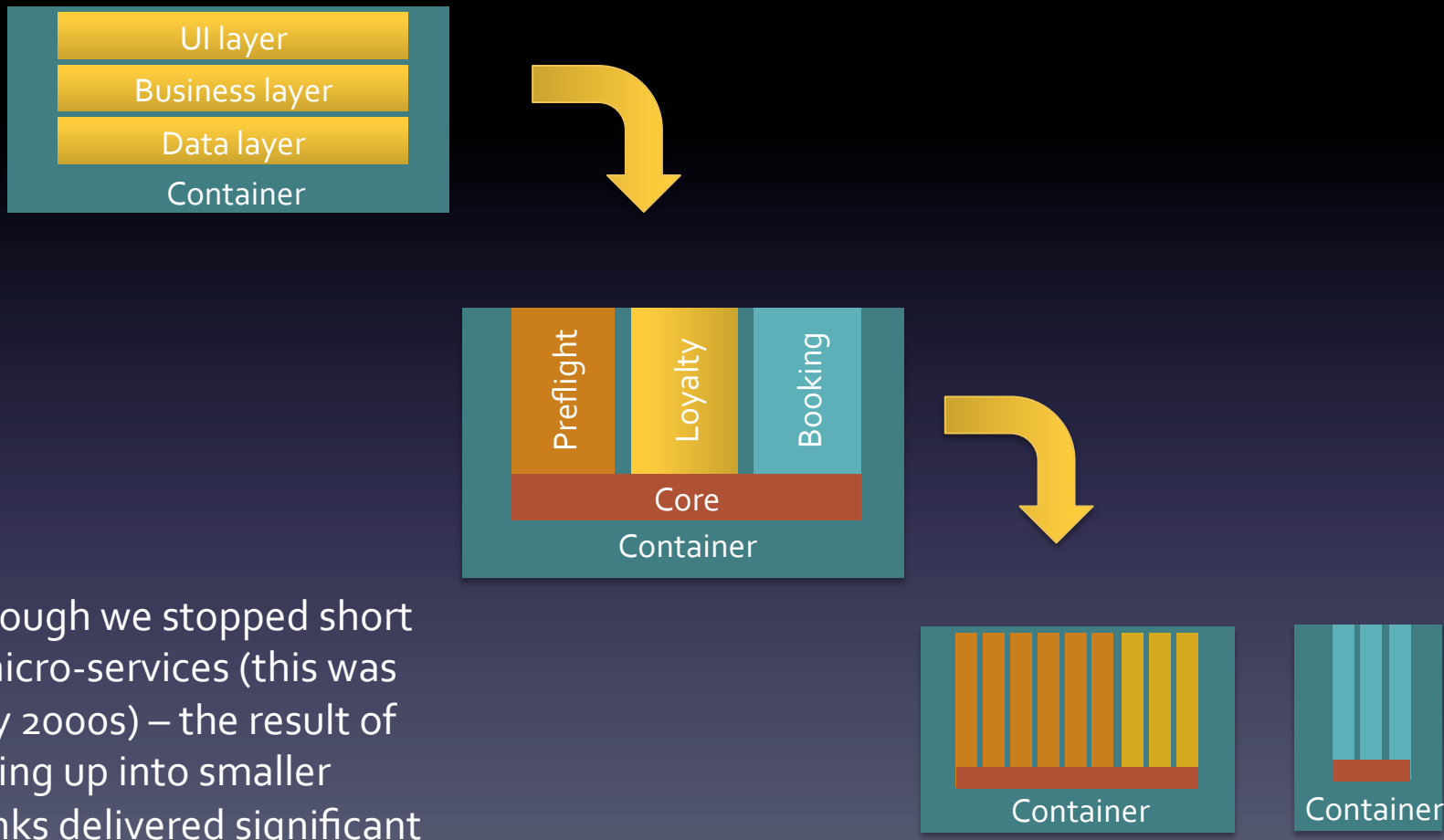


Hmm – looks like Netflix also seem to be moving away from monolithic application structures

Gilt too!


Enabling Microservice Architectures with Scala
by Kevin Scaldeferri on Jul 22, 2013
Summary
Kevin Scaldeferri reports on using Scala for an SBT plug-in, for unit and functional testing, type-safe shared configuration using Zookeeper, and live inventory with WebSocket and Akka Actors.
GETTING
(THE MICROSERVICES)
RELIGION
Bio
Kevin Scaldeferri works at Gilt Groupe, where he has worked on the core e-commerce and order processing components, as well as development tools.

Large airline website case study



Although we stopped short of micro-services (this was early 2000s) – the result of carving up into smaller chunks delivered significant value

What is a micro-service?

- Small (enough)
- Replaceable
- Independent
- Runs in its own O/S process (daemon)
- Easily Consumable (HTTP+JSON)
- Fast/Easy to startup
- Upgrade-able
- Free of Temporal coupling
- Encapsulation
- Forever young
- Valuable
- Testable
- Single Responsibility Principle

Complexity displacement

Many processes to manage

Increase in overall memory consumption

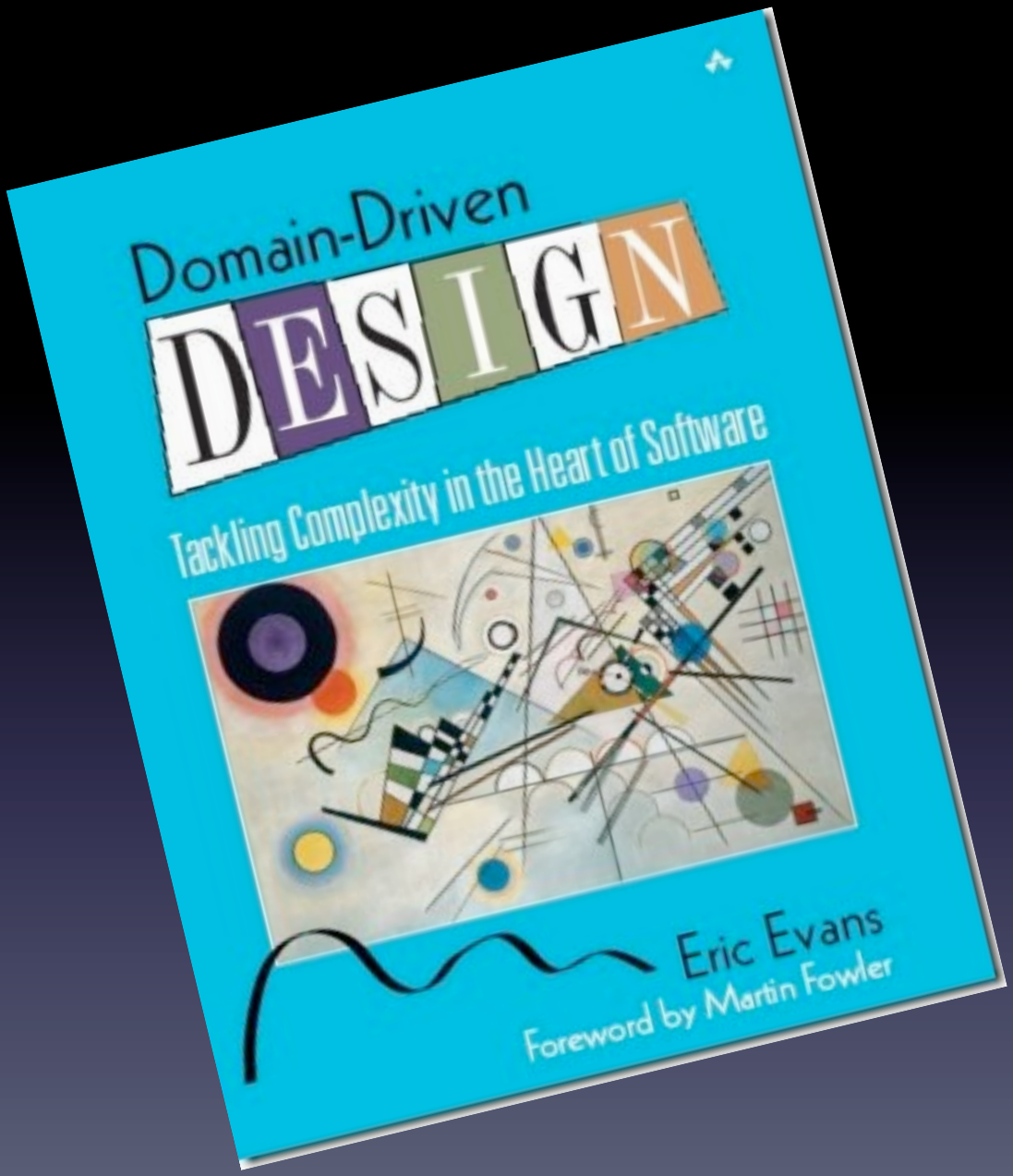
What are the costs of micro-services?

More moving parts

Operational complexity

Design considerations

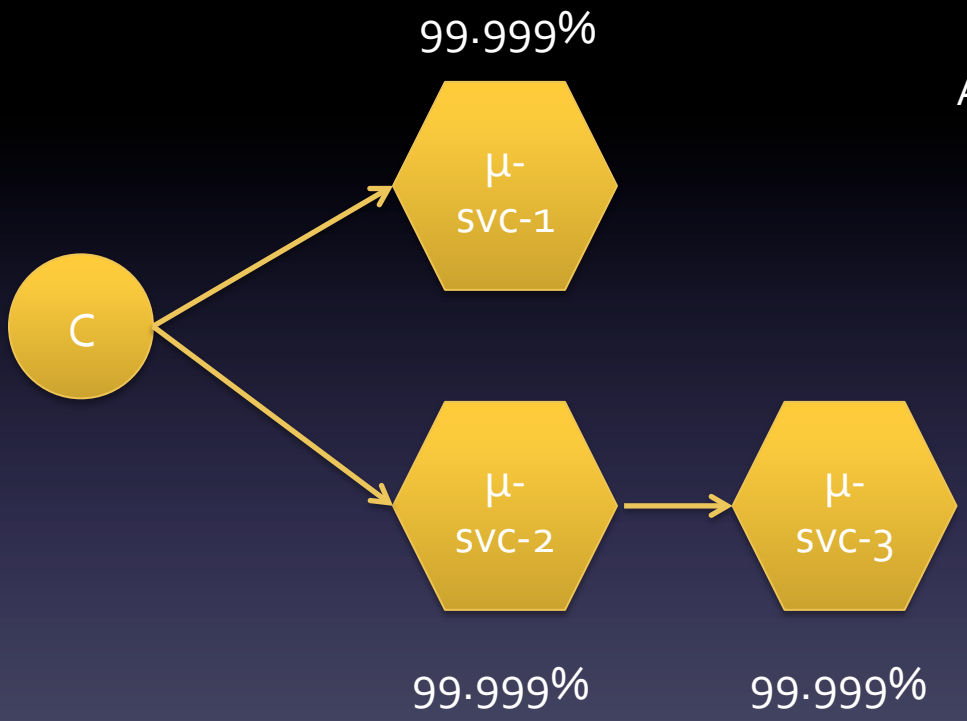
"Good OO design principles are equally applicable for the design of micro-services"



Domain-Driven Design (DDD) is complementary to a Micro-services based architecture

"The hardest part is going to be finding the seams in a monolithic application to form the boundaries of the micro-services"

Availability



$$\text{Availability} = \frac{\text{Uptime}}{\text{Uptime + Downtime}} \times 100\%$$

99.999% = 5.26 minutes downtime/year

Q. What is the overall availability? = A(μ-svc-1) X A(μ-svc-2) X A(A(μ-svc-3)

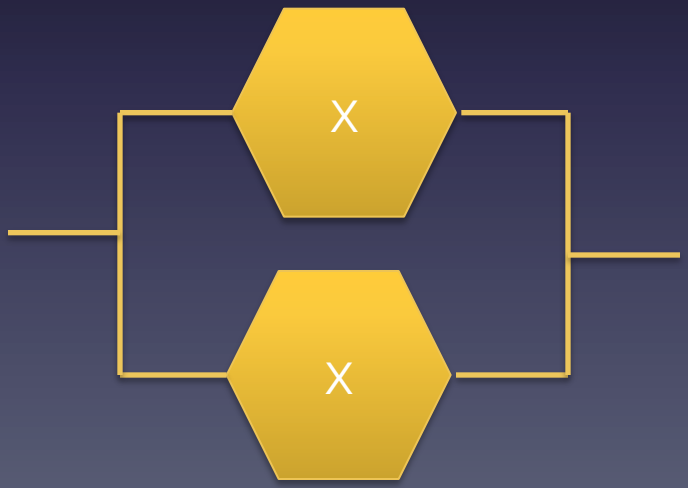
= 99.999% X 99.999% X 99.999%

= 99.997% = 15.77 minutes of downtime/year

Availability

$$A = A_x A_y$$

	X	Y	X & Y combined
	0.99	0.9999	0.989901
minutes/year	525600	525600	525600
Uptime mins/year	520344	525547.44	520291.9656
Downtime mins/year	5256	52.56	5308.0344
Downtime days/year	3.65	0.0365	3.686135

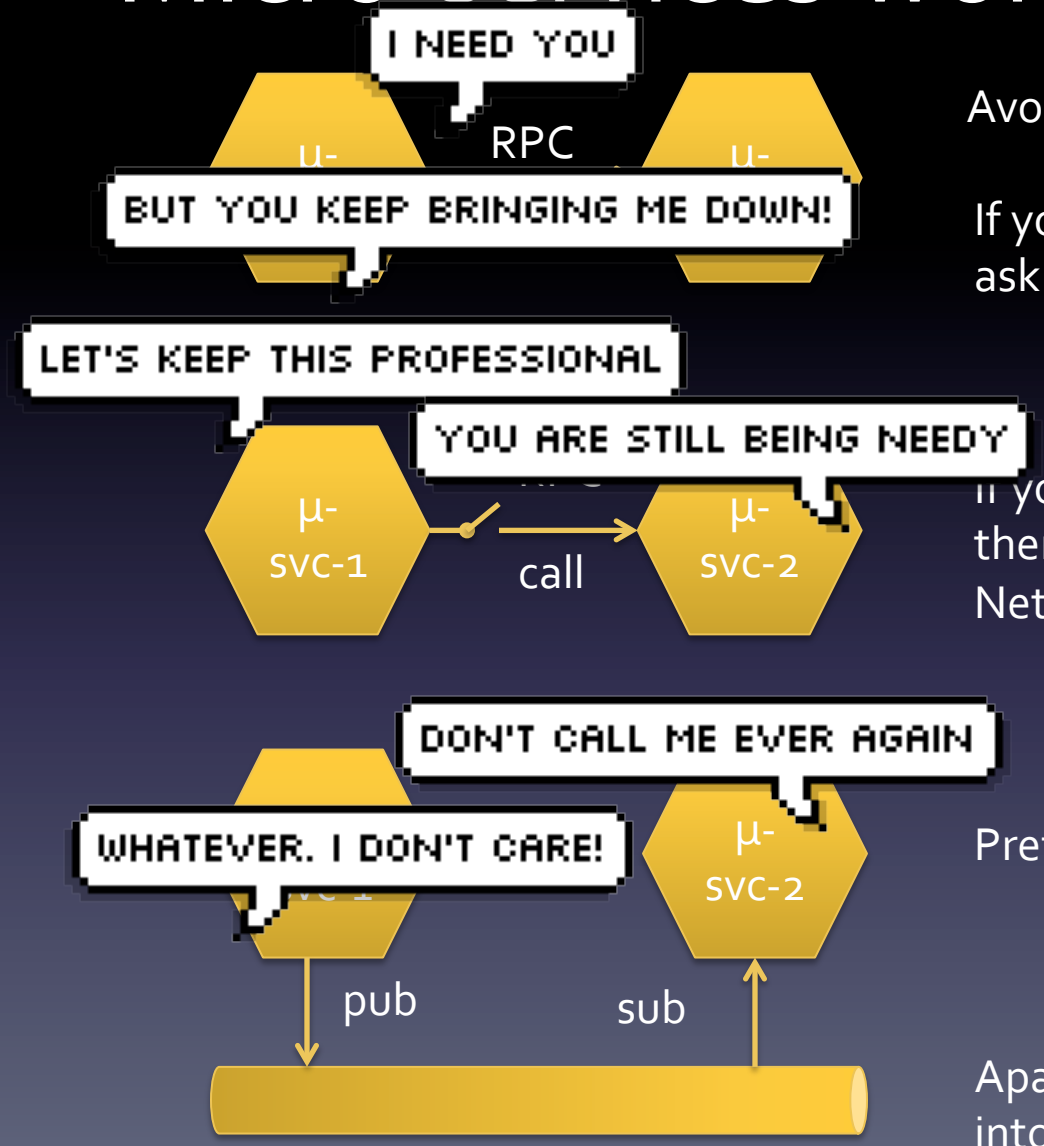


$$A = 1-(1-A_x)^2$$

	X	2x X	3x X
	0.99	0.9999	0.999999
minutes/year	525600	525600	525600
Uptime mins/year	520344	525547.44	525599.4744
Downtime seconds/year	315360	3153.6	31.536
Downtime mins/year	5256	52.56	0.5256
Downtime days/year	3.65	0.0365	0.000365

"Overall availability is reduced with each dependency. With micro-services the number of dependencies may increase."

Micro-services working together



Avoid synchronous RPC calls – why?

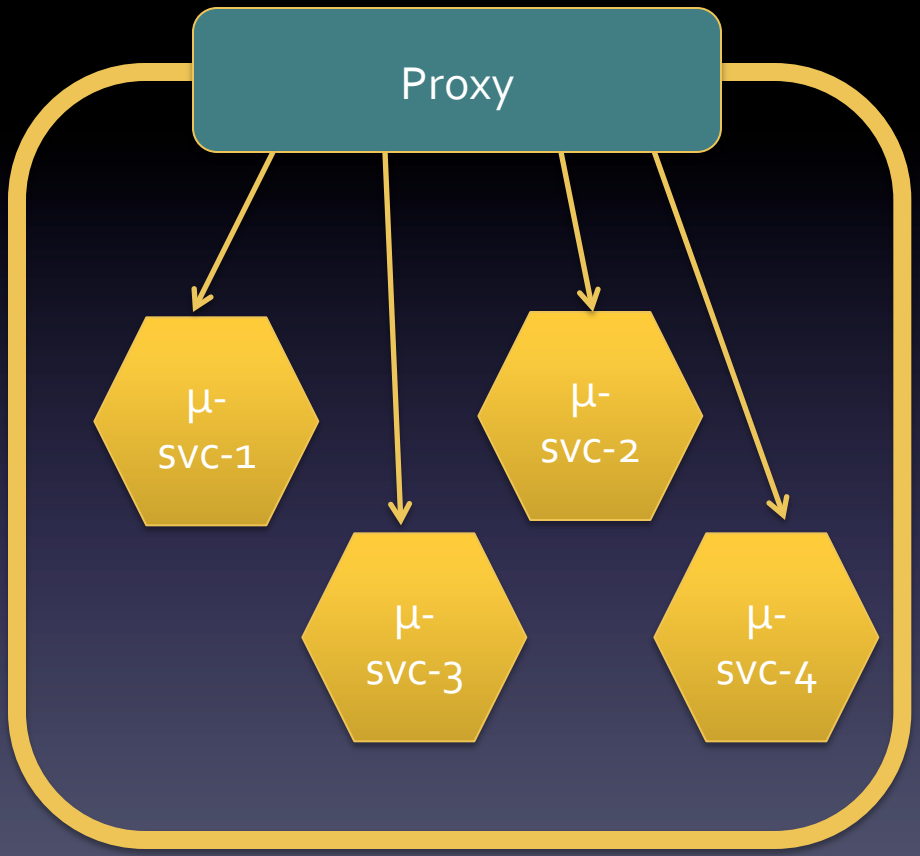
If you feel you need to do this, first ask – did I get the boundaries right?

If you still feel you need to do this, then consider a circuit breaker (e.g. Netflix Hystrix)

Prefer async pub/sub – why?

Apache Kafka may be worth looking into as a messaging backbone

Securing micro-services

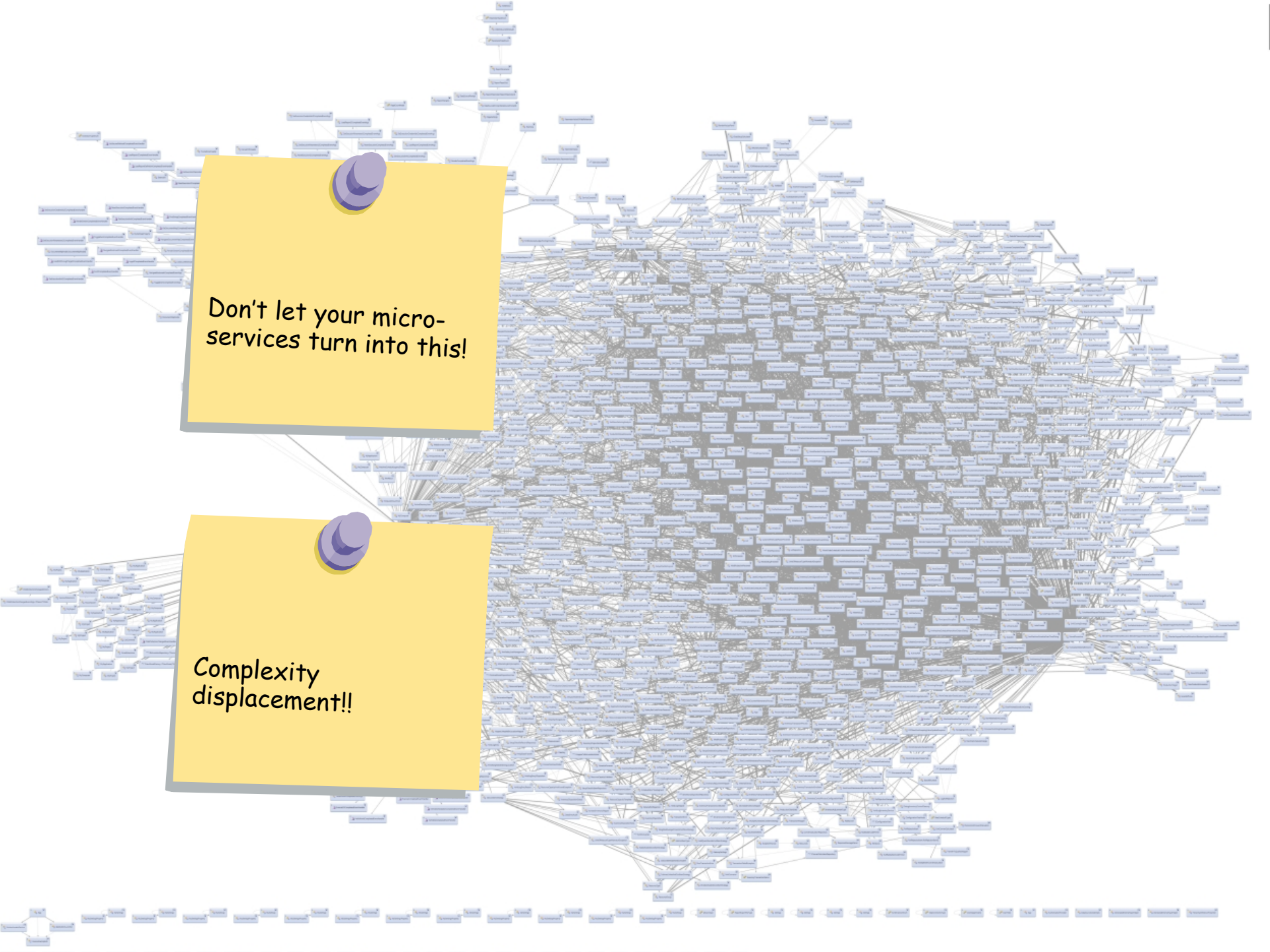


Consider fronting the micro-services with a proxy (with a circuit breaker)

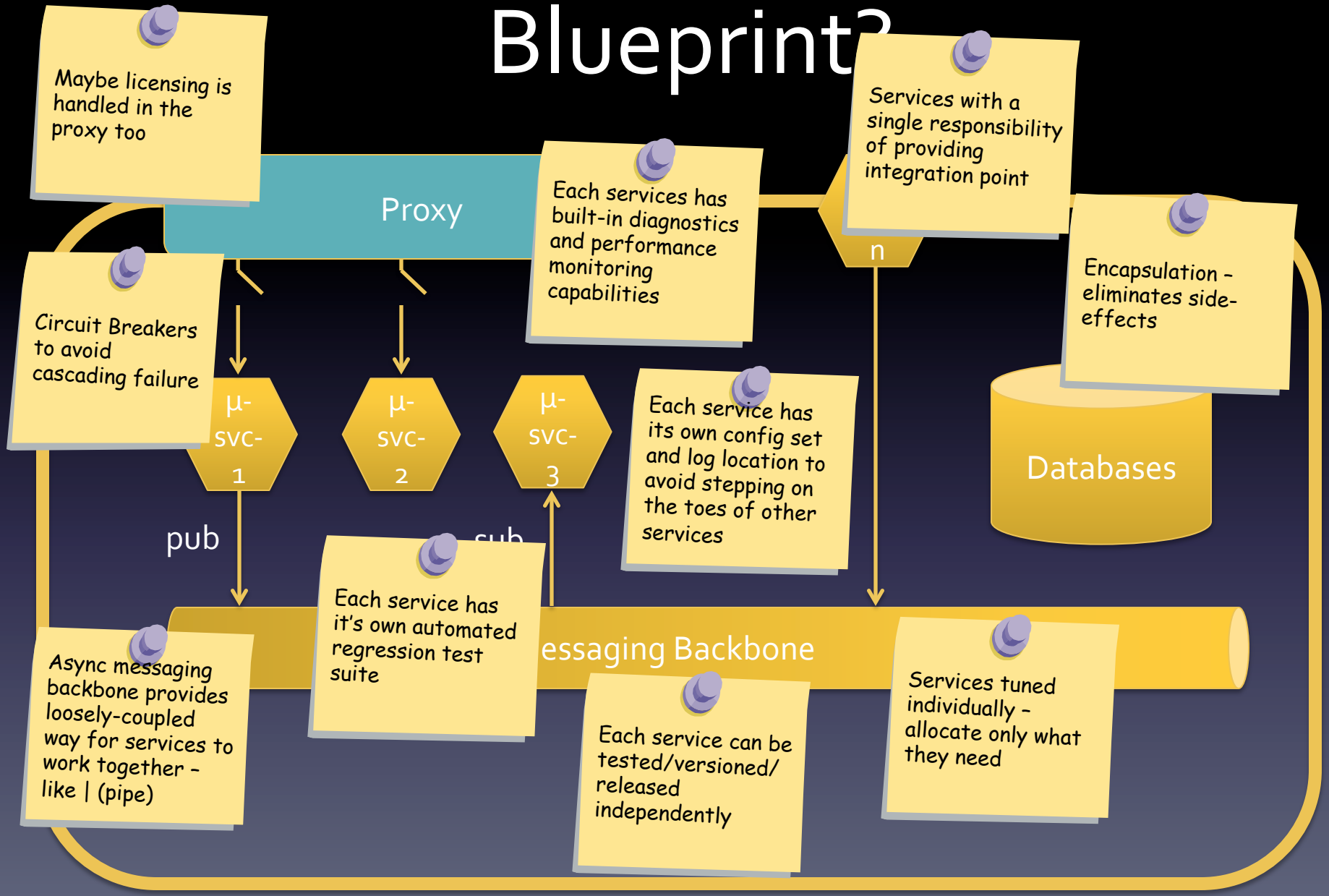
Maybe you want to control access

Maybe you need to invoke the services from a web client

Maybe you want to avoid having to open more holes in the firewall


Don't let your micro-services turn into this!
Complexity displacement!!

Blueprint?


Maybe licensing is handled in the proxy too
Proxy
Each services has built-in diagnostics and performance monitoring capabilities
Services with a single responsibility of providing integration point
n
Encapsulation - eliminates side-effects
Circuit Breakers to avoid cascading failure
µ-svc-1
µ-svc-2
µ-svc-3
Each service has its own config set and log location to avoid stepping on the toes of other services
Databases
pub
sub
Each service has it's own automated regression test suite
essaging Backbone
Async messaging backbone provides loosely-coupled way for services to work together - like | (pipe)
Each service can be tested/versioned/ released independently
Services tuned individually - allocate only what they need

Execution considerations

Use whatever language/ platform you are comfortable with today

For existing projects, carve off micro-services incrementally - avoid big-bang

.. as long as it meets "your" definition of what micro-services are

Polyglot (be careful of Polyglut)

Something like Dropwizard can help you get started quickly with a Java based implementation of micro-services

Dropwizard is a Java framework for developing ops-friendly, high-performance, RESTful web services.

Developed by Yammer to power their JVM-based backend services, Dropwizard pulls together **stable**, **mature** libraries from the Java ecosystem into a **simple**, **light-weight** package that lets you focus on *getting things done*.

Dropwizard has *out-of-the-box* support for sophisticated **configuration**, **application metrics**, **logging**, **operational tools**, and much more, allowing you and your team to ship a *production-quality* HTTP+JSON web service in the shortest time possible.

Getting Started » User Manual » About Dropwizard »

Something like this can help with performance monitoring

Metrics is a Java library which unparalleled insight into what does in production.

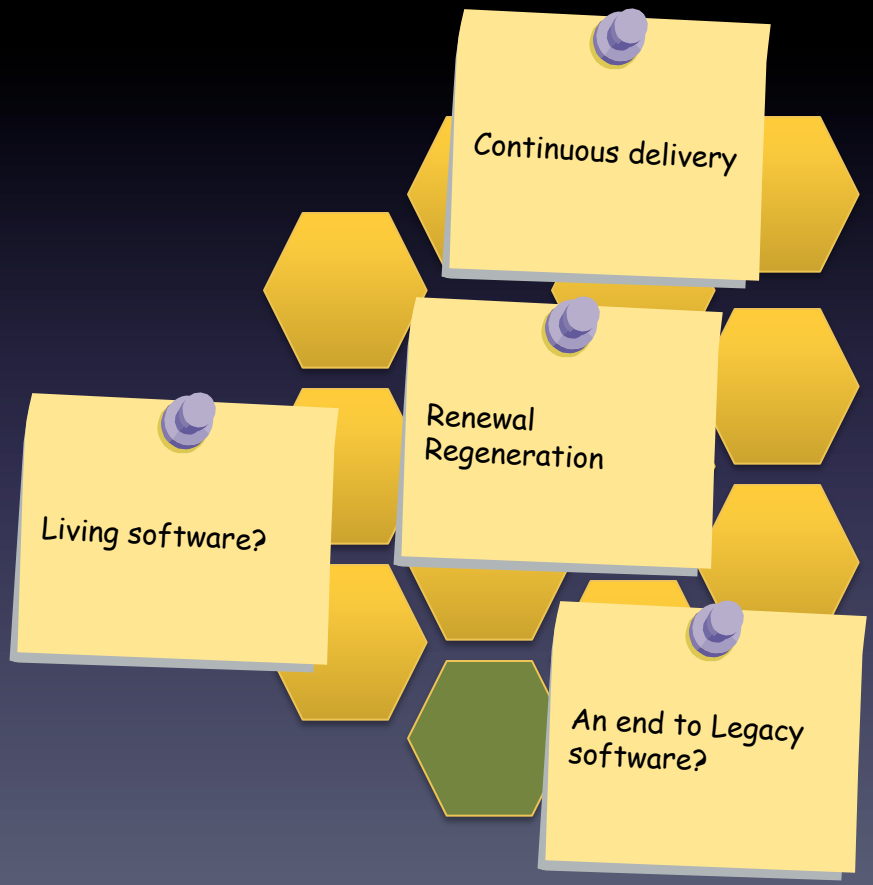
Metrics provides a powerful toolkit of ways to measure the behavior of critical components **in your production environment.**

With modules for common libraries like **Jetty**, **Logback**, **Log4j**, **Apache HttpClient**, **Ehcache**, **JDBI**, **Jersey** and reporting backends like **Ganglia** and **Graphite**, Metrics provides you with full-stack visibility.

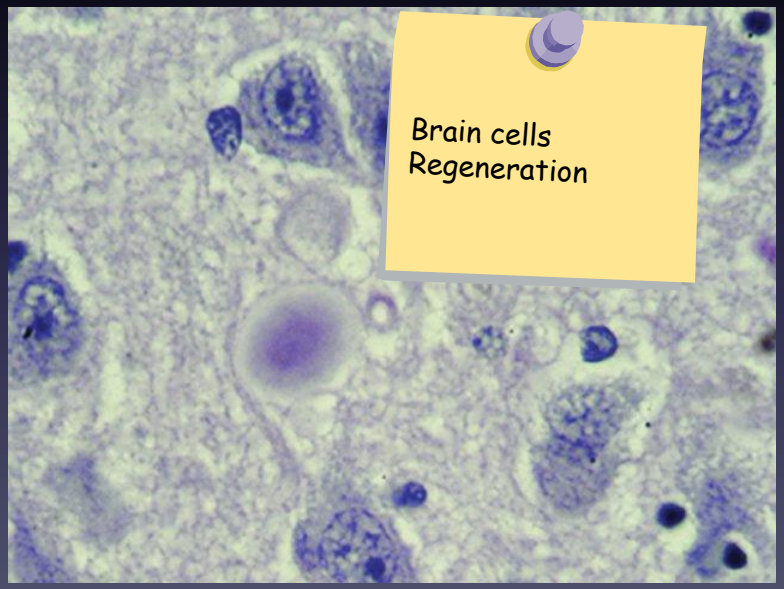
Getting Started » User Manual » About Metrics »

close()

Micro-services



Microbiology



http://www.genome.gov/dmd/img.cfm?node=Photos/Technology/Cells%20and%20biological%20pathways&id=79194

"If you look after the pennies, the pounds will look after themselves."

Lancope
VISION TO SECURE,
INTELLIGENCE TO PROTECT

Further Reading

- "The Second Coming of Java: A Relic Returns to Rule Web",
  Wired, Cade Metz, September 2013,
  http://www.wired.com/wiredenterprise/2013/09/the-second-coming-of-java/2/
- "Micro Services: Java, the Unix Way",
  QCon San Francisco 2012, James Lewis
  http://www.infoq.com/presentations/Micro-Services
- "GOTO Berlin: Microservices as an Alternative to Monoliths",
  InfoQ, Jan Stenberg, October 2013
  http://www.infoq.com/news/2013/10/goto-ber-microservices
- "Domain-Driven Design: Tackling Complexity in the Heart of Software",
  Addison-Wesley, Eric Evans, August 2003
  http://www.amazon.com/Domain-Driven-Design-Tackling-Complexity-Software/dp/0321125215
- "Baruco 2012: Micro-Service Architecture",
  Fred George, November 2012
  http://www.youtube.com/watch?v=2rKEveL55TY
- "Enabling Microservice Architectures with Scala ",
  Kevin Scaldeferri (Gilt), Jul 22 2013
  http://www.infoq.com/presentations/Microservice-Architectures-Scala
- "Adopting Continuous Delivery: Adjusting your Architecture",
  Rachel Laycock, Qcon,
  http://www.infoq.com/presentations/adopting-continuous-delivery
- http://yobriefca.se/blog/2013/04/29/micro-service-architecture/
- Dropwizard, http://dropwizard.codahale.com/
- Metrics, http://metrics.codahale.com/
- Netflix Hystrix, https://github.com/Netflix/Hystrix
- Apache Kafka, https://kafka.apache.org/
- System Reliability and Availability
  http://www.eventhelix.com/realtimemantra/faulthandling/system_reliability_availability.htm#.UtASOGRDuAR
- City of Atlanta Department of Watershed Management history http://www.atlantawatershed.org/inside-dwm/history/

DEVNEXUS

Lancope.
VISION TO SECURE,
INTELLIGENCE TO PROTECT

THANK YOU

Jason Chambers
Director of Engineering

DEVNEXUS