

3 Mini Talks of 10 minutes each



**The Seven Deadly Sins
of Microservices**
Daniel Bryant



**The Importance of Wise
Protocol-Layer Semantics
in the Design of
Microservices**
Arian Adair



HTTP Nanoservices
Michael Hendricks

QCon

THE SEVEN DEADLY SINS OF MICROSERVICES

DANIEL BRYANT

@DANIELBRYANTUK
CONTAINER SOLUTIONS LONDON



ContainerSolutions

DISCLAIMER

ANY RESEMBLANCE TO EXISTING PROJECTS
(WHETHER OVERRUNNING OR OVER-BUDGET)
IS PURELY COINCIDENTAL...



1. LUST - USING THE LATEST AND GREATEST TECH

01/06/15

@danielbryantuk

PICK YOUR (TECHNICAL) BATTLES...

- AS DAN MCKINLEY SAYS, “CHOOSE BORING TECHNOLOGY”
 - OPTIMIZE GLOBALLY ACROSS ORGANISATION
- MATT RAIBLE’S COMPARISON FRAMEWORK
 - OBJECTIVITY(?)
 - DOCUMENT DECISIONS

Criteria	Struts 2	Spring MVC	Wicket	JSF 2	Tapestry	Stripes	GWT	Grails	Rails	Flex	Vaadin	Lift	Play
Developer Productivity	0.50	0.50	0.50	0.50	1.00	0.50	1.00	1.00	1.00	0.00	1.00	0.50	1.00
Developer Perception	0.50	1.00	1.00	0.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Learning Curve	1.00	1.00	0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00
Project Health	0.50	1.00	1.00	1.00	0.50	0.50	1.00	1.00	1.00	0.50	1.00	1.00	1.00
Developer Availability	0.50	1.00	0.50	1.00	1.00	0.50	1.00	0.50	1.00	1.00	1.00	0.50	0.00
Job Trends	1.00	1.00	0.50	0.50	0.50	0.50	1.00	0.00	1.00	1.00	1.00	0.00	0.50
Tooling	1.00	1.00	1.00	0.50	1.00	1.00	0.50	1.00	1.00	0.50	0.50	0.50	0.50
Components	0.00	0.00	1.00	1.00	1.00	0.00	0.50	0.50	0.50	1.00	1.00	0.00	0.00
Ajax	0.50	1.00	0.50	0.50	0.50	0.50	1.00	0.50	0.50	0.50	1.00	1.00	0.50
Plugins or Add-Ons	0.50	0.00	1.00	1.00	0.50	0.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00
Scalability	1.00	1.00	0.50	0.50	0.50	1.00	1.00	0.50	0.50	0.50	0.50	0.50	1.00
Testing	1.00	1.00	0.50	0.50	1.00	1.00	0.50	1.00	1.00	0.00	0.50	0.50	1.00
i18n and t10n	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.50	0.50	1.00	1.00
Validation	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50
Multi-language Support (Groovy / Scala)	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00
Quality of Documentation/Tutorials	0.50	1.00	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Books Published	1.00	1.00	0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00	0.50	0.50	0.00
RESTful API (client and server)	0.50	1.00	0.50	0.50	0.50	0.50	0.50	1.00	1.00	0.50	0.50	0.50	0.50
Mobile / iPhone Support	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Degree of Risk	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50
Totals	14.5	17	15	13.5	15	14	17	17	17.5	17	13.5	15.5	11.5



2. GLUTTONY - EXCESSIVE **COMMUNICATION** PROTOCOLS

01/06/15

@danielbryantuk

STANDARDIZE ACROSS THE ORGANIZATION

- CHOOSE **ONE** SYNCHRONOUS PROTOCOL
 - E.G. JSON OVER HTTP
- CHOOSE **ONE** ASYNCHRONOUS PROTOCOL
 - E.G. RABBITMQ

DON'T GOLD-PLATE, BUT KNOW YOUR OPTIONS: PROTOBUF, THRIFT, ZEROMQ, MQTT



3. GREED - ALL YOUR SERVICE ARE **BELONG TO US**

CONWAY WAS TELLING THE TRUTH... DEAL WITH IT!



01/06/15

@danielbryantuk



4. SLOTH - CREATING A **DISTRIBUTED MONOLITH**

CAN'T DEPLOY SERVICES INDEPENDENTLY?

- CHECK YOUR 'BOUNDED CONTEXTS'
- CONSUMER-DRIVEN CONTRACTS FTW
 - MARTIN FOWLER: [CDC - A SERVICE EVOLUTION PATTERN](#)

“CDC IS BDD FOR MICROSERVICES”

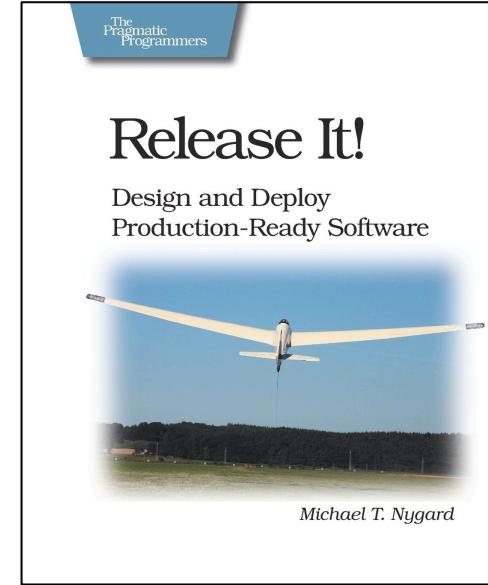


5. WRATH - BLOWING UP WHEN BAD THINGS HAPPEN

01/06/15

@danielbryantuk

BRING IN THE MONKEYS (OR MICHAEL NYGARD)





6. ENVY - THE SHARED SINGLE DOMAIN FALLACY

ONE MODEL TO RULE THEM ALL...

- ONE MODEL
 - BREAKS ENCAPSULATION
 - INTRODUCES COUPLING
- KNOW YOUR DDD
 - ENTITIES
 - VALUE OBJECTS
 - AGGREGATES AND ROOTS





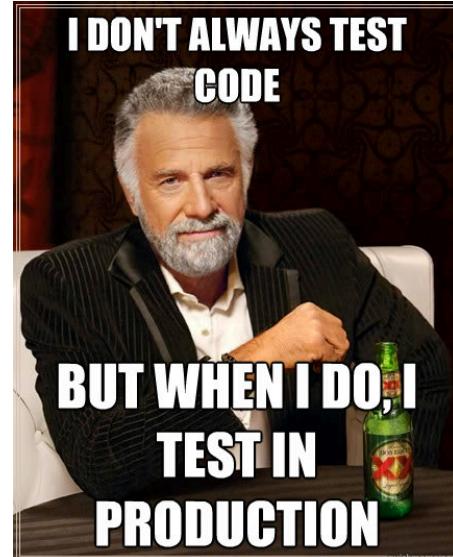
7. PRIDE - TESTING IN THE WORLD OF TRANSIENCE

01/06/15

@danielbryantuk

TESTING WITH MICROSERVICES IS DIFFICULT...

- INVEST IN YOUR **BUILD PIPELINE TESTING**
 - SERENITY BDD
 - WIREMOCK / SABOTEUR
 - JENKINS PERFORMANCE PLUGIN
- DO IT IN **PRODUCTION(?)**
 - NETFLIX
 - GILT



THANKS...

@DANIELBRYANTUK

DANIEL.BRYANT@CONTAINER-SOLUTIONS.COM

@CONTAINERSOLUTI

WWW.CONTAINER-SOLUTIONS.COM

(CREDIT TO TAREQ ABEDRABBO FOR INSPIRATION/GUIDANCE)

facebook

The Importance of Wise Protocol Semantics in the Design of Microservices

Arian Adair
QConNY June 2015
 @_silversupreme

computer



distributed systems 😱

microservices === distributed systems

How To Make Distributed Systems
that Don't Require a Crystal Ball to
Use.

Three Principles of Stress-Free Service Communication.

1. Your Transport Layer Matters
More than You Think it Does.

HTTP is Not Your Friend.

"Every HTTP API eventually grows until it ~~can consume email~~ contains a buggy implementation of about half of TCP."

TCP is your Frenemy.

Pretend You're In Third Grade and
Pass Messages Around.

(Yes, it's OK to build an RPC
framework on top of this.)

**2. Use a Structured Data Format
with a Schema.**

Data Formats

Or, how I Learned to Stop Worrying and Love Thrift

```
struct Task {  
    1: i64 task_id  
    2: string title  
    3: string description  
    4: i64 uid  
    5: bool completed  
}
```

Data Formats

Or, how I Learned to Stop Worrying and Love Thrift

```
struct Comment {  
    1: i64 id  
    2: i64 taskID  
    3: i64 uid  
    4: string text  
    5: i64 created  
}
```

Data Formats

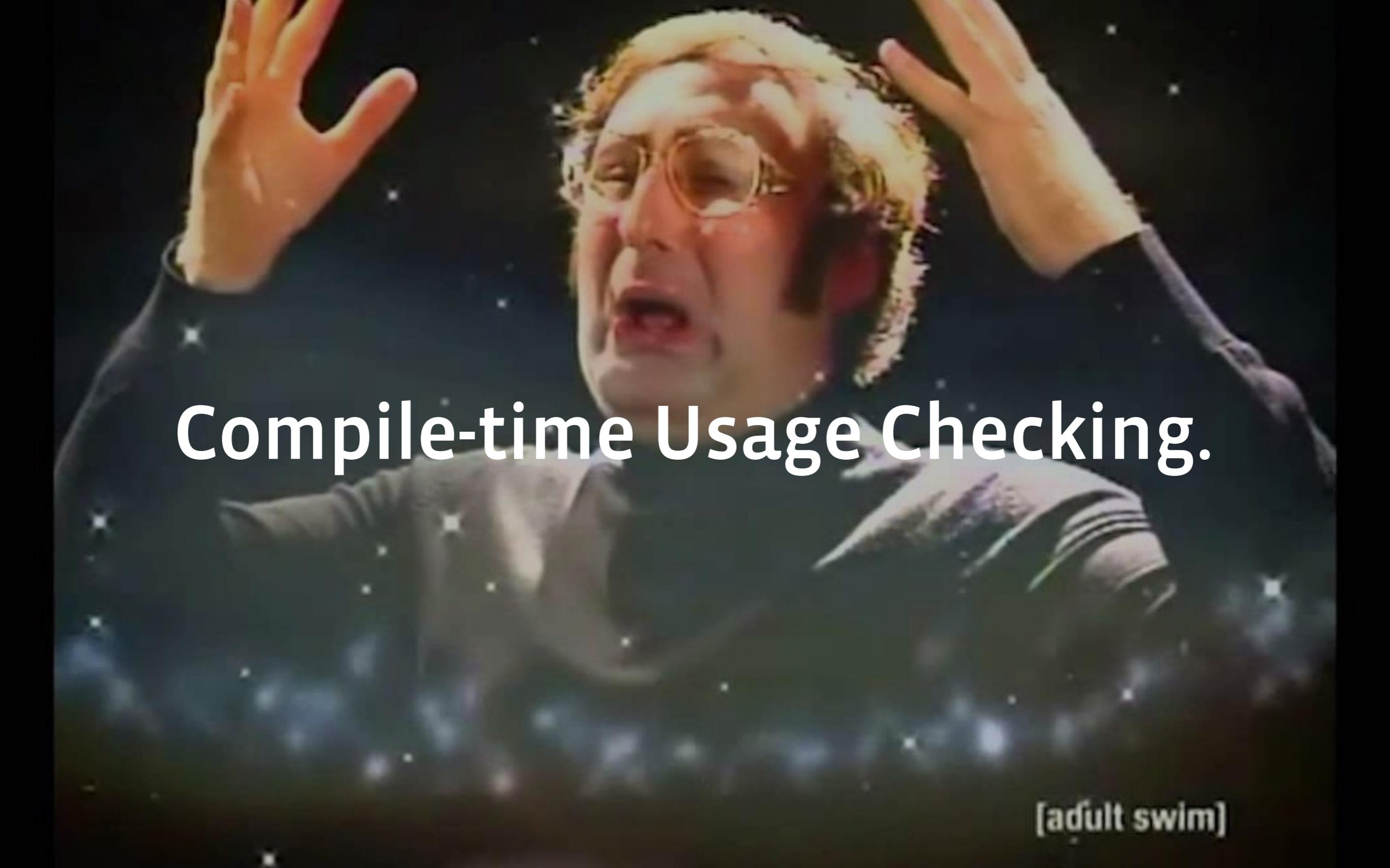
Or, how I Learned to Stop Worrying and Love Thrift

```
service TaskService {  
    // Pass in a mostly-filled Task struct, returns ID of the new Task.  
    i64 createTask(1:Task t)  
}
```

Data Formats

Or, how I Learned to Stop Worrying and Love Thrift

```
service TaskService {  
    // Get all the comments on a particular task.  
    list<Comment> getCommentsForTask(1:i64 taskID)  
}
```

A man wearing a white spacesuit and a helmet with a clear visor is floating in a dark, star-filled space. He has a shocked or surprised expression, with his mouth open and hands raised in a 'stop' or 'freeze' gesture. The background consists of numerous small white stars.

Compile-time Usage Checking.

[adult swim]

**3. Smarter People than You or I
have Already Built What You Need.**

Open-Source Transport/Protocol Formats

It's OK to Stand on the Shoulders of Giants

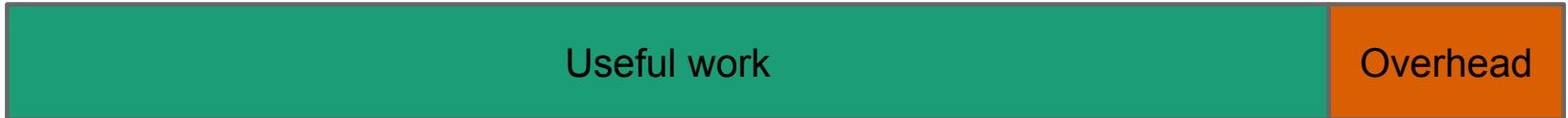
Name	Creator	License
Thrift	Facebook	Apache
gRPC/Protocol Buffers	Google	BSD
Cap'n Proto	Sandstorm	MIT

**"There are three types of lies: lies,
damn lies, and consistency
guarantees."**



@_silversupreme

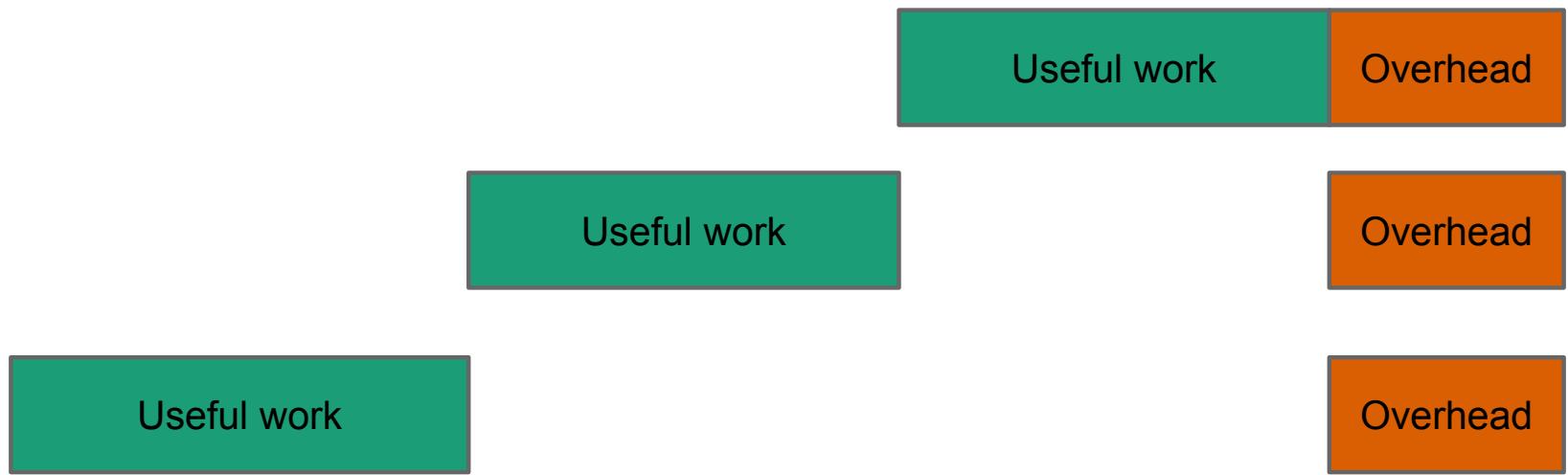
HTTP Nanoservices



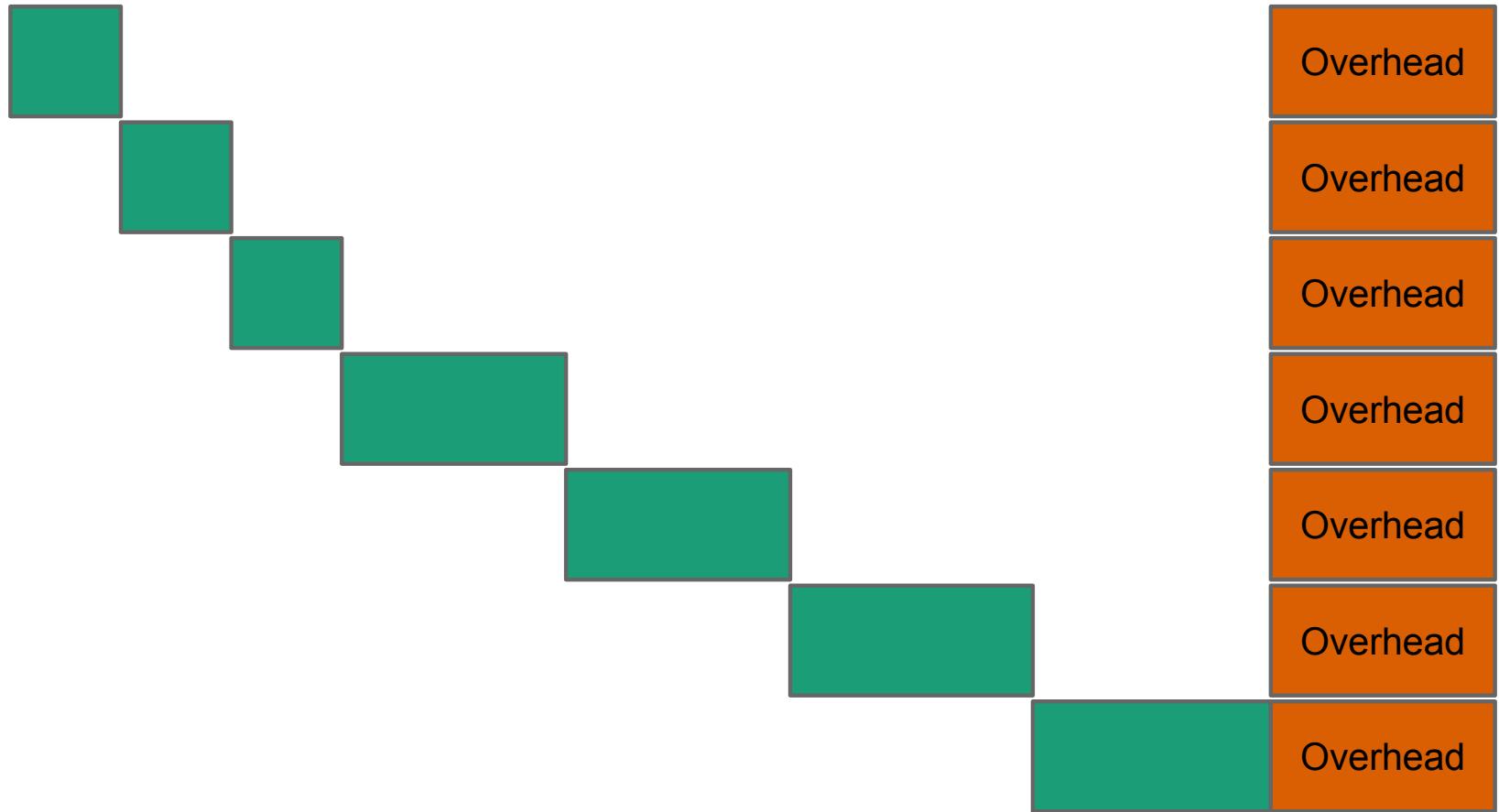
Useful work

Overhead

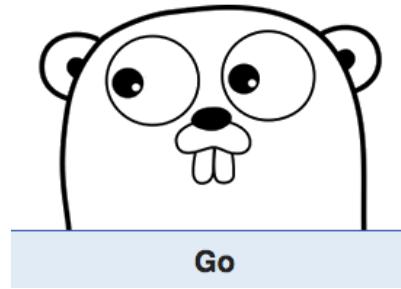
Monolithic service



Microservices



Nanoservices



SWI Prolog

>> Haskell

Polyglot

hookscript

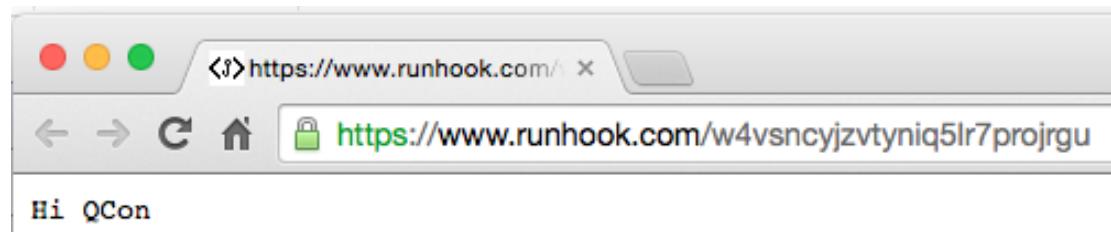
Select Language: **Perl**

```
1 use Hookscript;
2 say "Hi QCon";
3 |
```

Hi QCon

Save

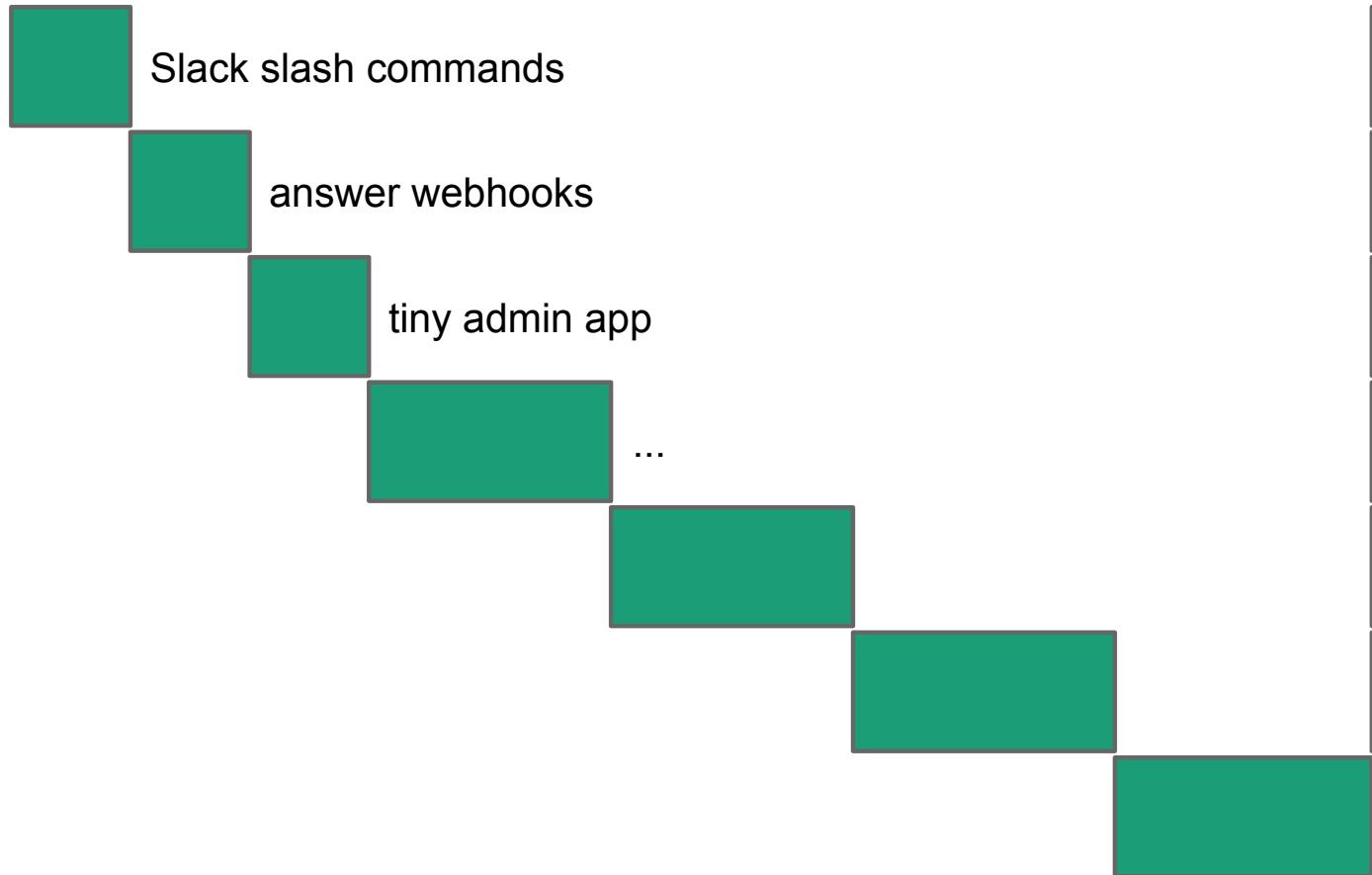
Paste code ...



... get a server

Servers can

- access full HTTP request
- make outbound network requests
- store persistent data
- run arbitrary code or libraries



on Hookscript

hookscript.com

thanks