

Dynamic vs. Static: Having a Discussion without Sounding Like a Lunatic

David Greenberg



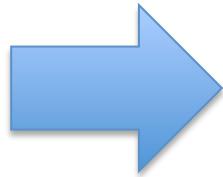
Two Sigma

LambdaConf 2015



Who am I?

- Designed Piplin, Clojure on FPGAs



Who am I?

- Designed Piplin, Clojure on FPGAs
- Architected Two Sigma compute cluster



Who am I?

- Designed Piplin, Clojure on FPGAs
- Architected Two Sigma compute cluster
- Couch + scotch + paper = Enjoy



Disclaimer

This talk is about how to choose a programming language, and how to discuss that choice with others.

Disclaimer

It's about how types affect that choice.

Disclaimer

I'd like to outrage everyone today,
equally.

Why [not] use a Dynamic Language?

Section I

Who's visited a website today?

Powered by



JavaScript

So, we're done. JavaScript won.

Let's talk about JSON

JSON

The greatest data interchange format

Great

adjective – of ability, quality, or
eminence considerably above the
normal or average

Great

adjective – of ability, quality, or
eminence considerably above the
normal or average

JSON – Well Specified

json key order problems



Web

Videos

Shopping

News

Images

More ▾

Search tools

About 947,000 results (0.34 seconds)

[java - Keep the order of the JSON keys during JSON ...](#)

stackoverflow.com/.../keep-the-order-of-the-json-keys-during-json-conv... ▾

Dec 23, 2010 - I am using the **JSON** library provided here

<http://www.json.org/java/index.html> to convert a **json** string I have to CSV. But the problem I have is, ...

[java - When JsonObject's keys are iterated they aren't in the ...](#)

stackoverflow.com/.../when-jsonobjects-keys-are-iterated-they-arent-in-t... ▾

Aug 13, 2011 - I converted it to **JSON** object and then get the **keys** and iterate it. The problem is that when I iterate it isn't in the same **order** as in response from ...

JSON – Rich Representation

I've seen so many different standards for JSON date format:

"\"\\Date(1335205592410)\\\""

.Net JavaScr

"\"\\Date(1335205592410-0500)\\\""

.Net DataCon

"2012-04-23T18:25:43.511Z"

JavaScript b

"2012-04-21T18:25:43-05:00"

ISO8601

Which one is the right one? Or best? Is there any sort of standard

JSON - Efficient

json efficiency

Web

News

Images

Videos

Shopping

More ▾

Search tools

About 466,000 results (0.55 seconds)

[BSON - Wikipedia, the free encyclopedia](#)

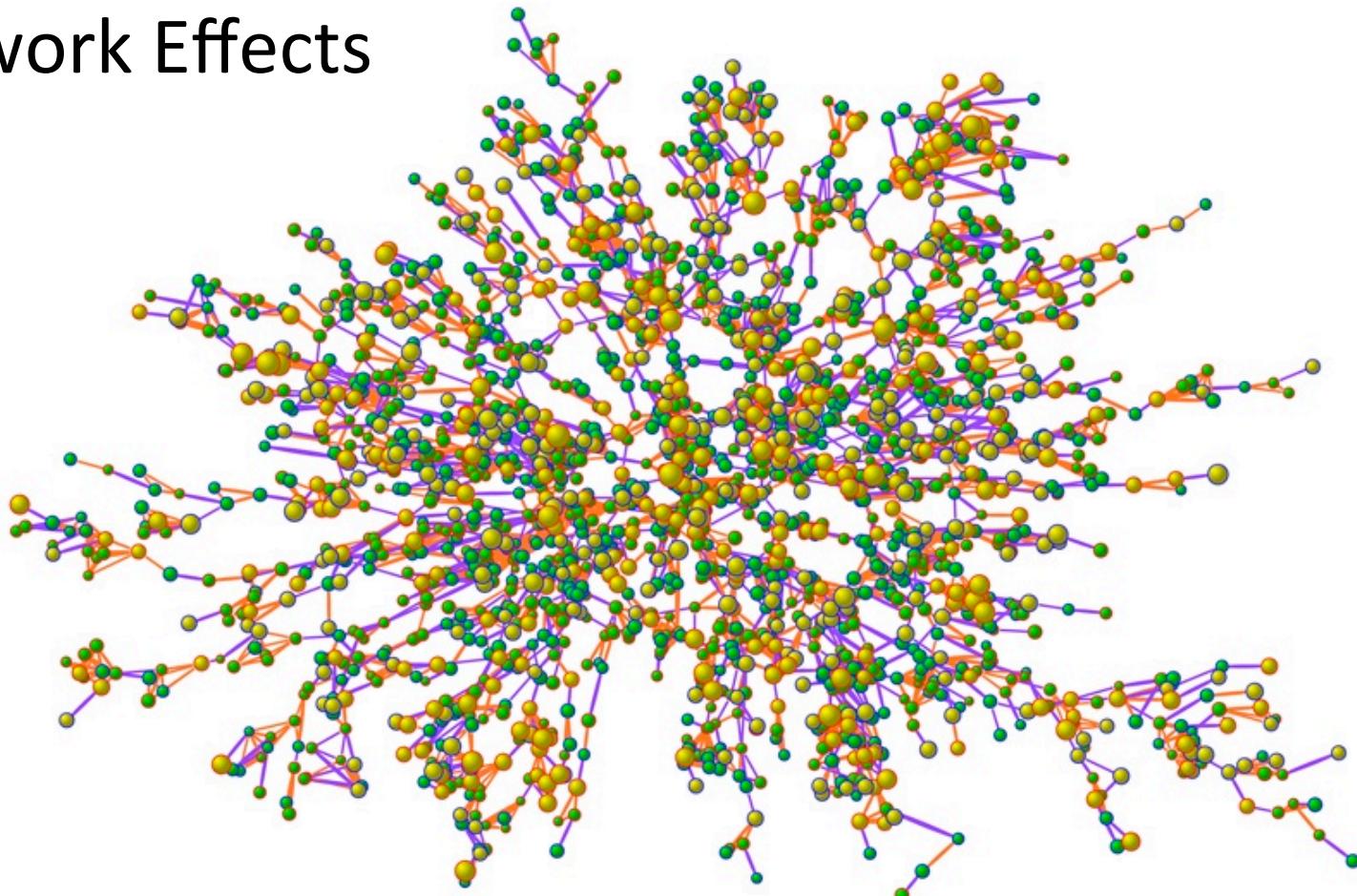
[en.wikipedia.org/wiki/BSON](#) ▾ [Wikipedia](#) ▾

Compared to **JSON**, BSON is designed to be **efficient** both in storage space and scan-speed. Large elements in a BSON document are prefixed with a length ...

[Data types and syntax](#) - [Efficiency](#) - [See also](#) - [References](#)

Why is it so popular?

Network Effects



Why is it so popular?

```
// Parsing JSON in JavaScript  
  
var obj = JSON.parse(text);
```

Why is it so popular?

```
# Parsing JSON in Python
```

```
import json
```

```
data = json.loads(json_data)
```

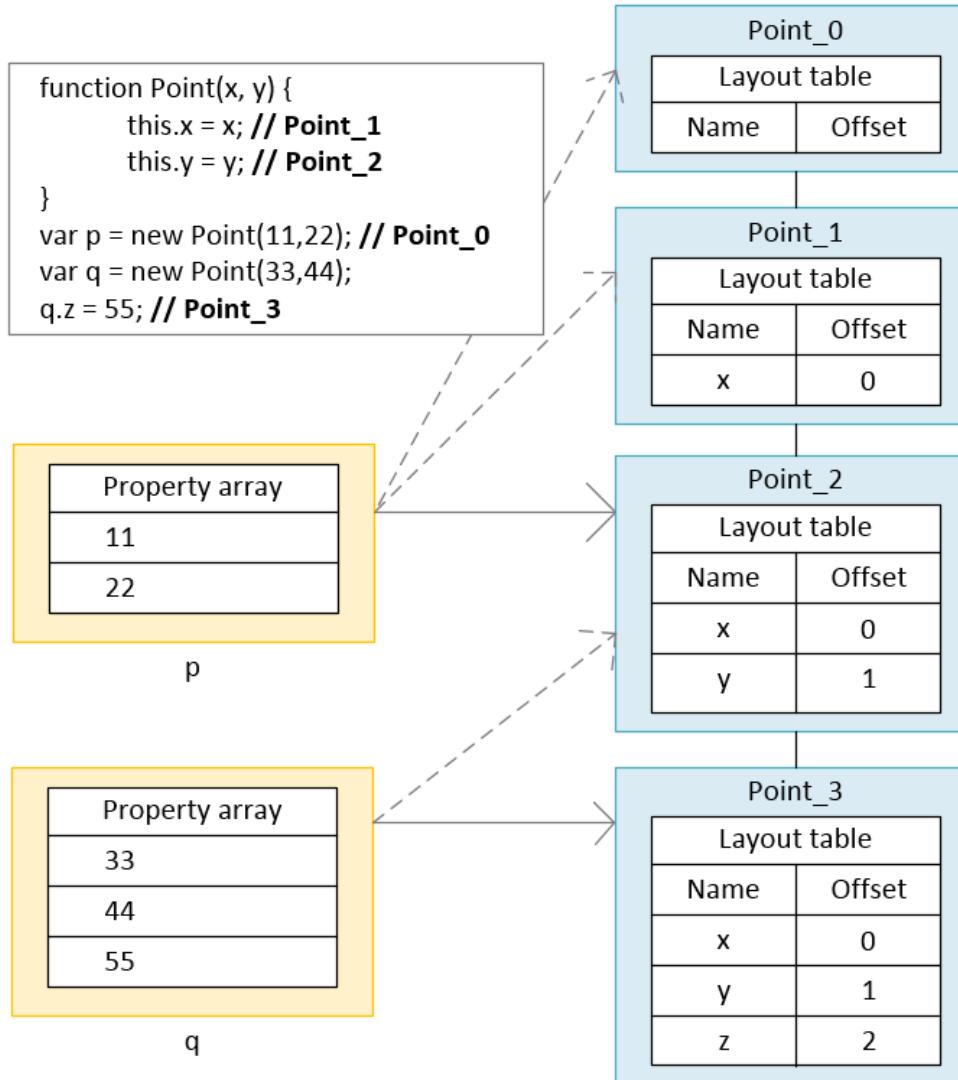
Why is it so popular?

```
// Parsing JSON in Java  
  
import org.json.*;  
  
JSONObject obj = new JSONObject(" . . . ");
```

Why is it so popular?

- Compatibility and ease-of-use
- Schemalessness makes for easy parsing

JSON's Hidden Types



JSON's Hidden Types

```
function Point(x, y) {  
    this.x = x; // Point_1  
    this.y = y; // Point_2  
}  
var p = new Point(11,22); // Point_0  
var q = new Point(33,44);  
q.z = 55; // Point_3
```

JSON's Hidden Types

Property array

11

22

p

JSON's Hidden Types

```
function Point(x, y) {  
    this.x = x; // Point_1  
    this.y = y; // Point_2  
}  
var p = new Point(11,22); // Point_0  
var q = new Point(33,44);  
q.z = 55; // Point_3
```

JSON's Hidden Types

Point_0

Layout table

Name

Offset

JSON's Hidden Types

```
function Point(x, y) {  
    this.x = x; // Point_1  
    this.y = y; // Point_2  
}  
var p = new Point(11,22); // Point_0  
var q = new Point(33,44);  
q.z = 55; // Point_3
```

JSON's Hidden Types

Point_1

Layout table

Name	Offset
x	0

JSON's Hidden Types

Property array

11

22

p

JSON's Hidden Types

```
function Point(x, y) {  
    this.x = x; // Point_1  
    this.y = y; // Point_2  
}  
var p = new Point(11,22); // Point_0  
var q = new Point(33,44);  
q.z = 55; // Point_3
```

JSON's Hidden Types

Point_2	
Layout table	
Name	Offset
x	0
y	1

JSON's Hidden Types

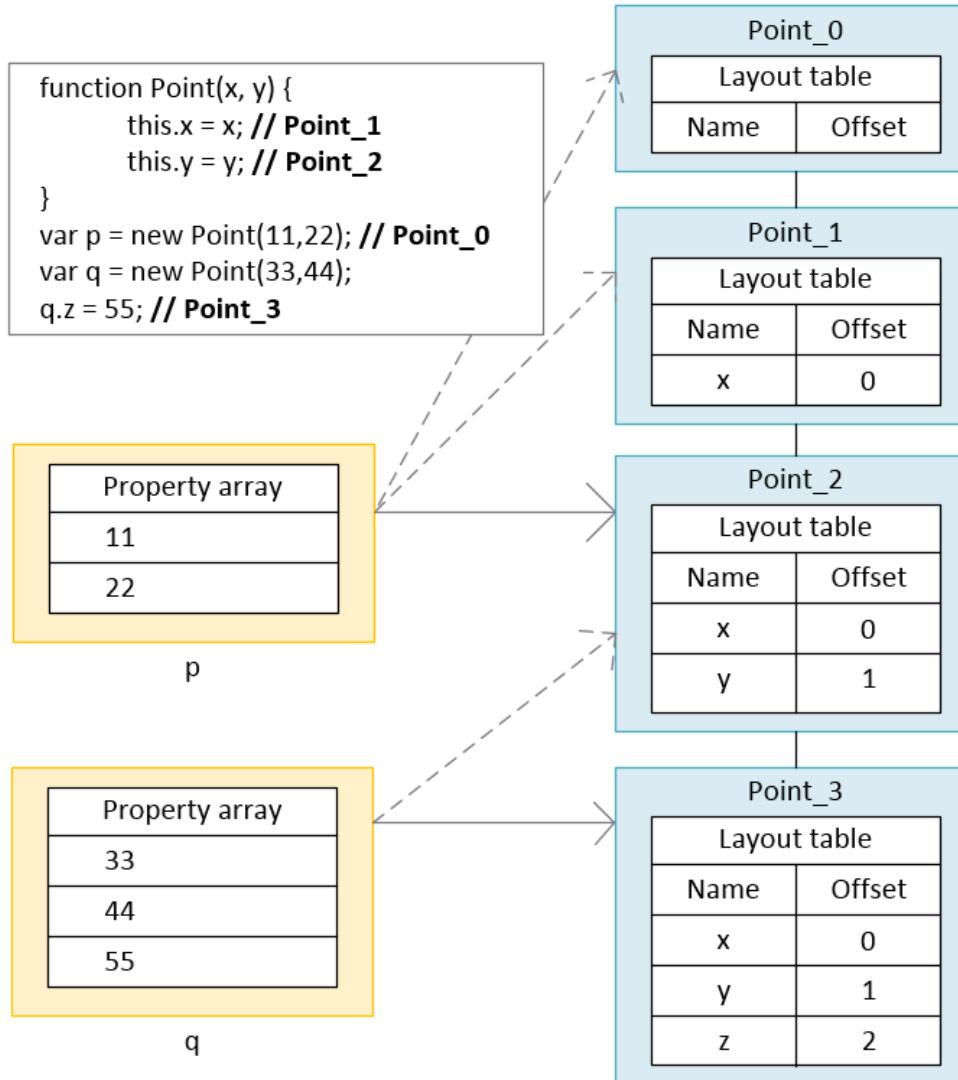
Property array

11

22

p

JSON's Hidden Types



JSON's Hidden Types

```
function Point(x, y) {  
    this.x = x; // Point_1  
    this.y = y; // Point_2  
}  
var p = new Point(11,22); // Point_0  
var q = new Point(33,44);  
q.z = 55; // Point_3
```

JSON's Hidden Types

Point_3	
Layout table	
Name	Offset
x	0
y	1
z	2

JSON's Hidden Types

Property array

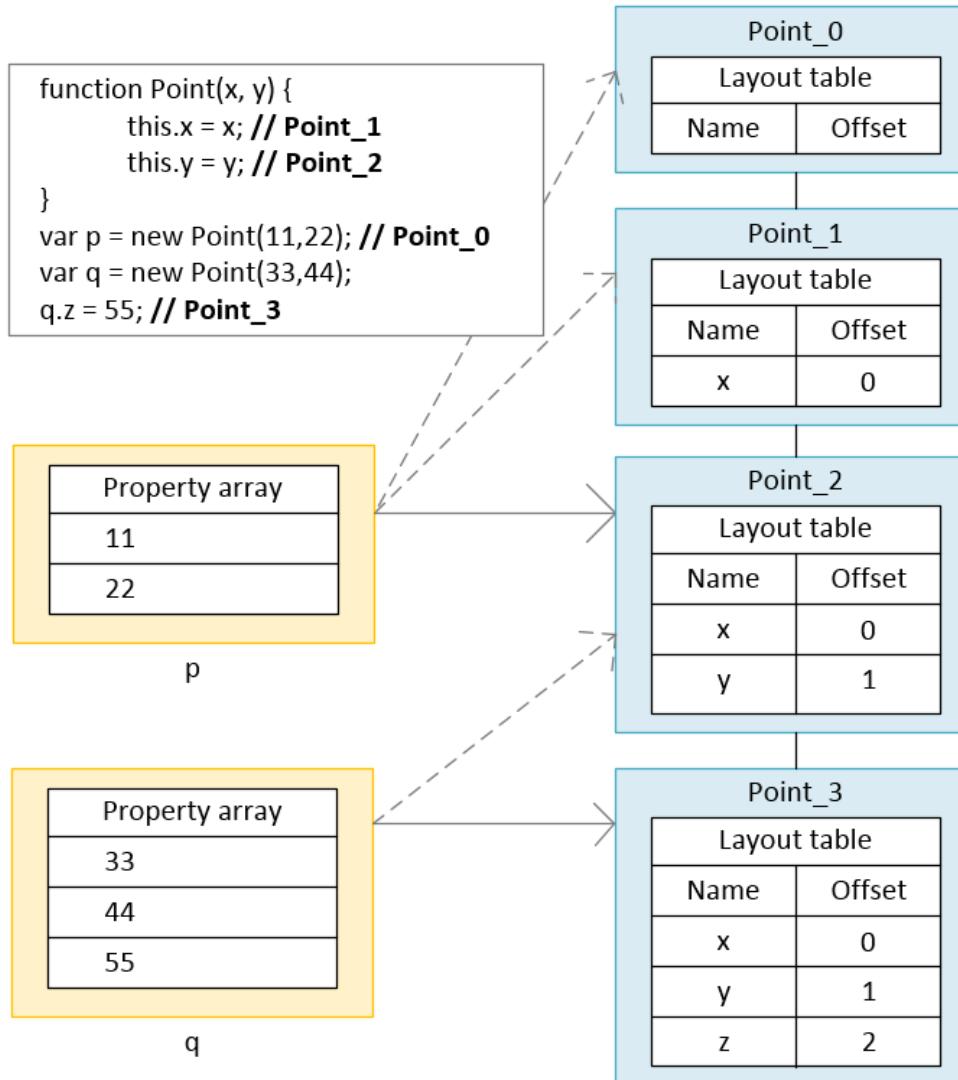
33

44

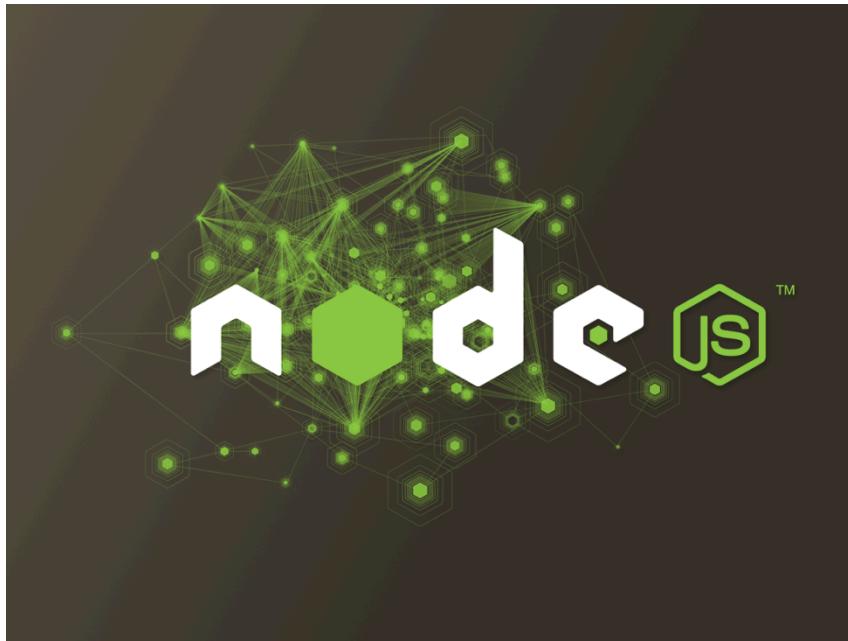
55

q

JSON's Hidden Types



There's something there



- Node.JS at Paypal
- Moved from Java
- Faster execution
- Fewer bugs

It's not about the language

This just matters in a vacuum

It's about the engineers

And their relationship to the machine

It's about the engineers

And their happiness

It's about the engineers

And their passion

It's about the engineers

And how much they care

Github

- Valued at \$750M in 2012
- Guess what language it is?
- Ruby and Erlang
 - Seriously?



Problem with Ruby

- Monkey Patching
 - Powerful tool
- You can change functionality without forking the framework



But then I did this...

```
irb:0> "hello".to_s
=> "hello"
irb:0> class String
irb:1>   def to_s
irb:2>     'fail'
irb:2>   end
irb:1> end
=> nil
irb:0> "hello".to_s
=> "fail"
```

Monkey patching in Haskell

We just call it adding a new
instance to a typeclass

What are we really talking about?

The Expression Problem

	List.add	List.get	List.clear	List.size
ArrayList				
LinkedList				
Stack		Existing method implementations		
Vector				
Your new class here		Your new method implementations		

Ruby never had static type safety

So did it solve the Expression
Problem?

How to advocate for languages, poorly

Section II

The World's a Scary Place



The Ethics of Typesystems

The aforementioned JavaScript
developers are unethical, and should
have their licenses revoked

Paul Snively

- Respected in Scala community
- Blogs about type systems
- Spoke at StrangeLoop 2012 & 2014
 - Types vs Tests: An Epic Battle
 - Type Systems: The Good, Bad, and Ugly

Paul Snively



Katherine Fellows

@kf



+ Follow

"That's why a lot of crappy Ruby developers are now crappy Clojure developers."

Shots fired at the type theory talk!

#strangeloop

St Louis, MO



...

The ~~Ethics~~ of Typesystems

~~The aforementioned JavaScript
developers are unethical, and should
have their licenses revoked~~

The Emotions of Typesystems

~~The aforementioned JavaScript
developers are unethical, and should
have their licenses revoked~~

The Emotions of Typesystems

I refuse to believe that my beloved
preference isn't the best choice

Beer Snobs



What Should Drive Our Decisions?

Section III

Business Needs



Why are we even paid?



Sell Shoes

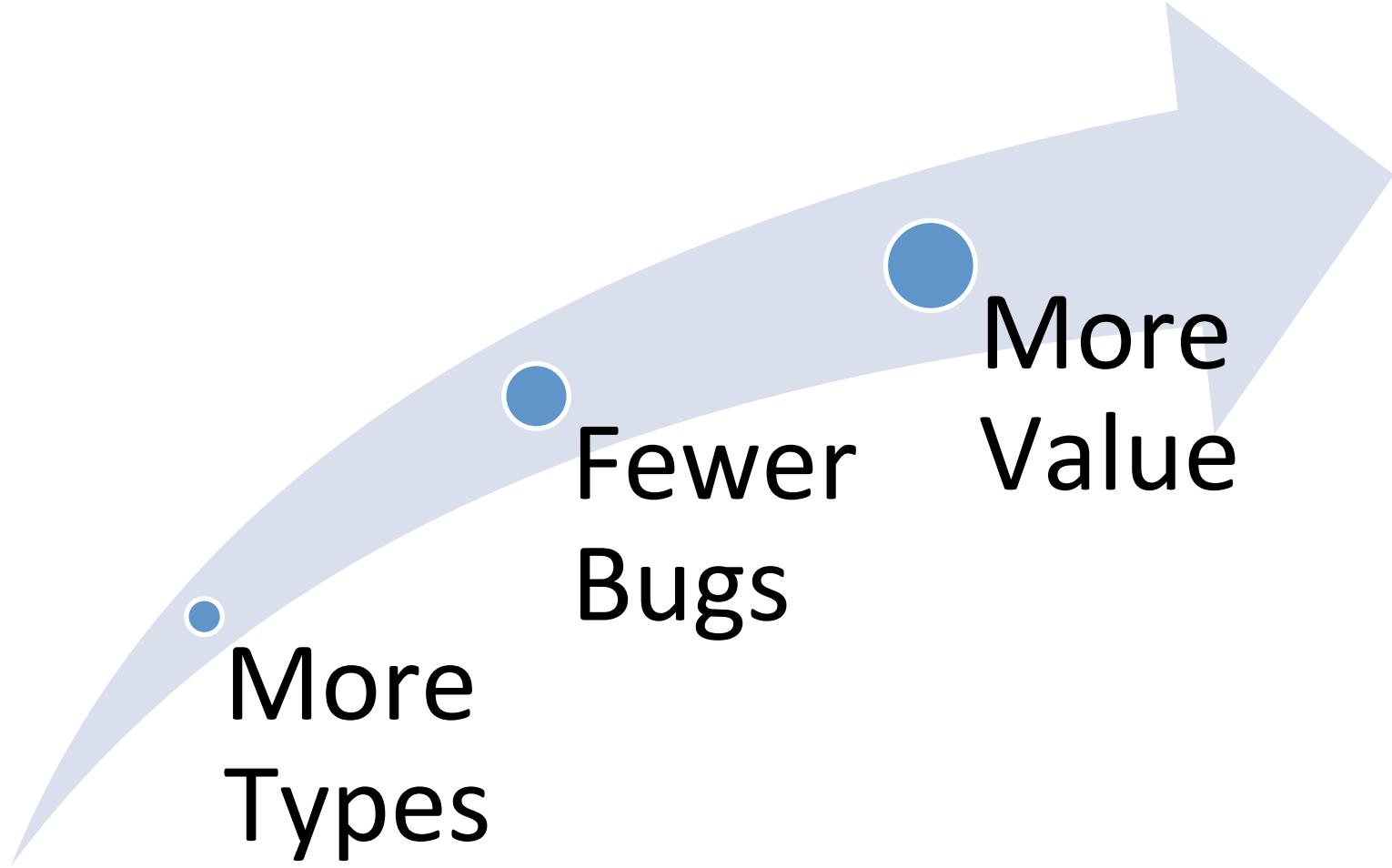


Sell laptops

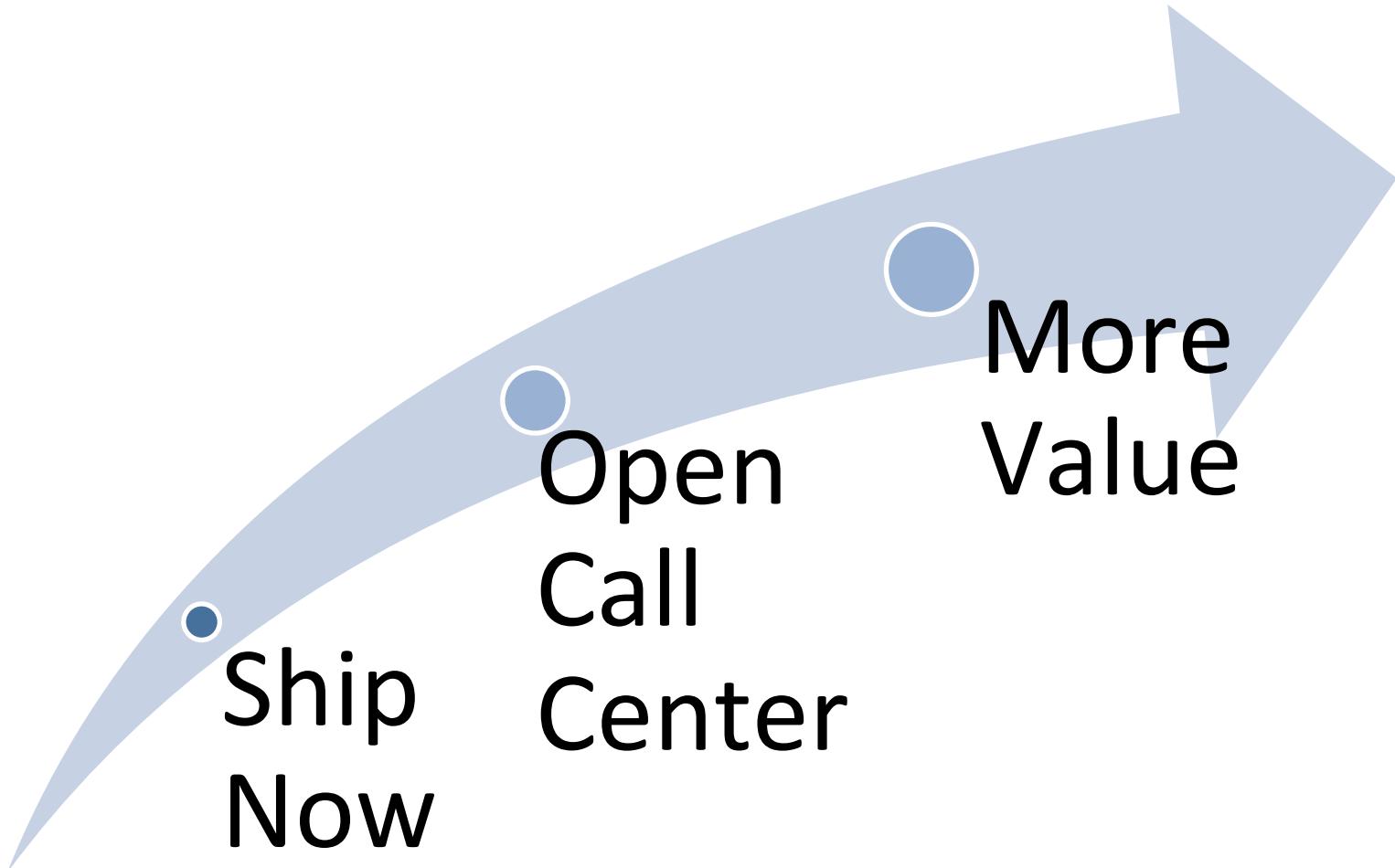


Sell compute infrastructure

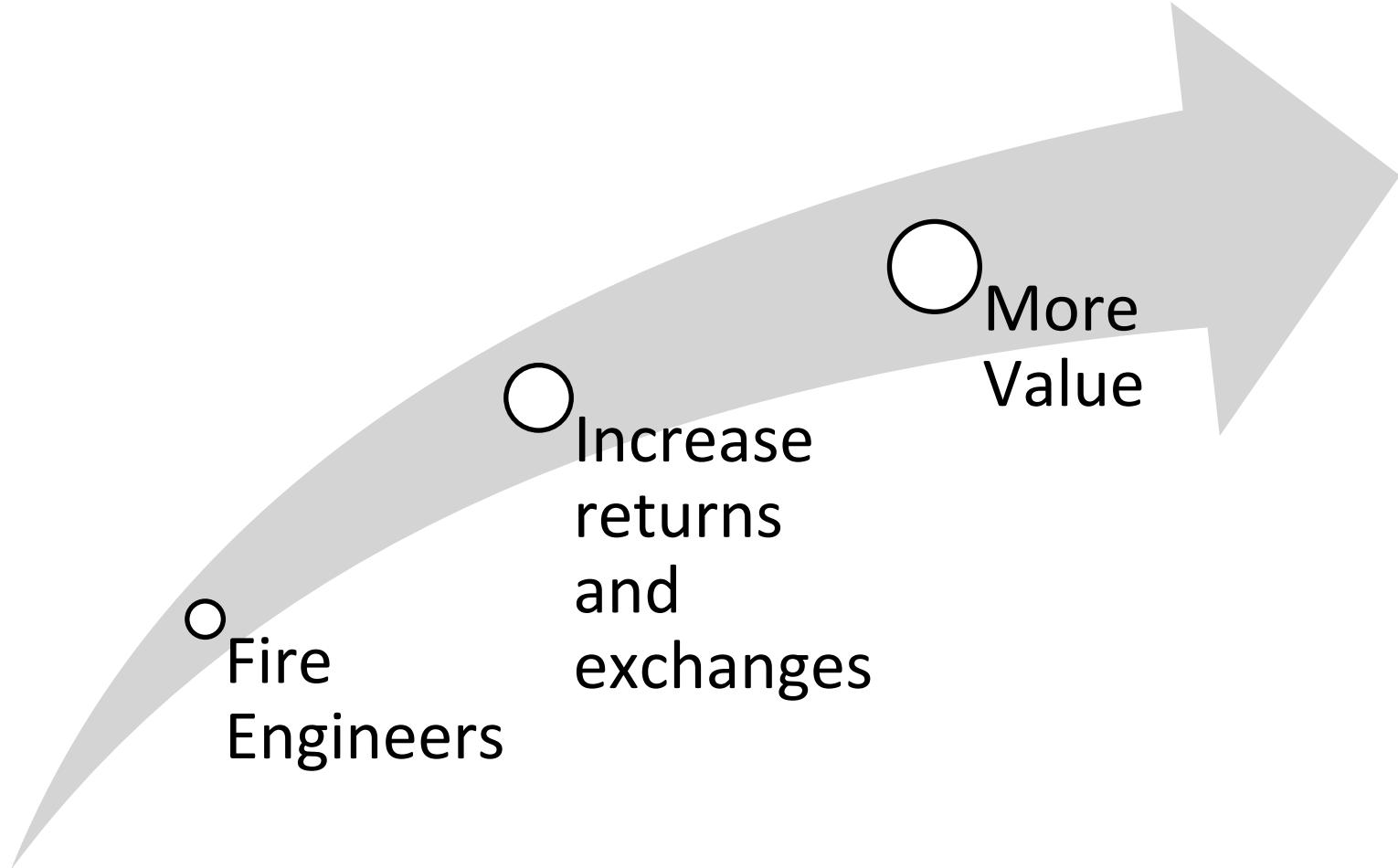
The Plan



The Plan



The Plan



We're screwed

But I still think types are great!

Let's make a distinction

Who is your type system for?

Who is your type system for?

The Computer

- Java
- C
- C++
- Go

The Human

- OCaml
- Haskell

But Java helps me reduce bugs!

Does the type system prove global
properties of your system?

But Java helps me reduce bugs!

Does a type-checked program
guarantee zero runtime errors?

Who are you talking to?

- Some developers have only used computer-typed languages
- Can't say "types" and expect them to understand!
- First, create the distinction

So when should we use typed languages?

Type systems like Haskell and OCaml

Critical Systems



Databases

Aerospace



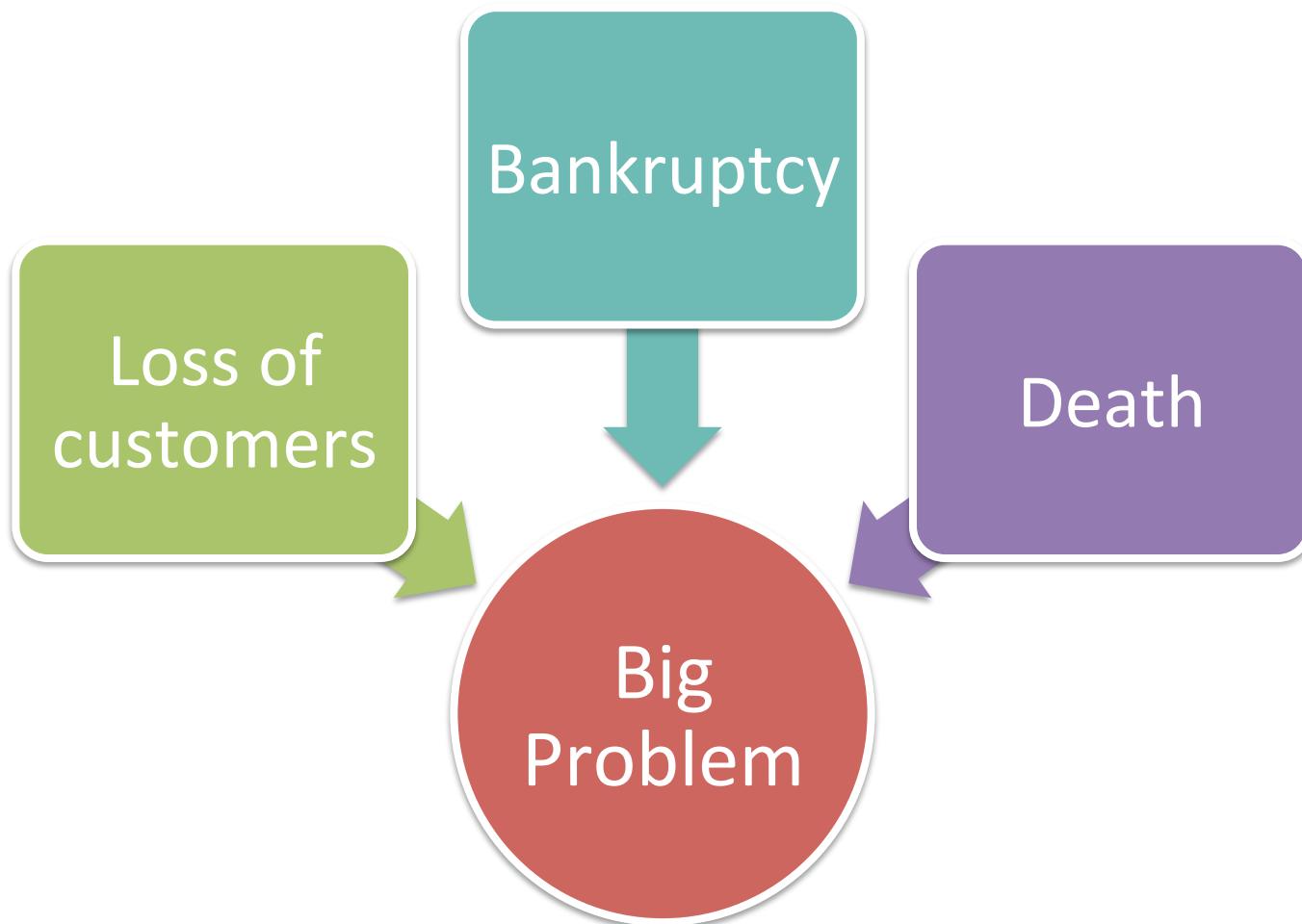
Transportation



Energy



Impact of Issues



How do we trust our software?

- Design review
- Code review
- Unit testing
- Integration testing
- Static analysis
- Theorem Proving

How do we trust our software?

- Design review
- Code review
- Unit testing
- Integration testing
- Static analysis
- Theorem Proving

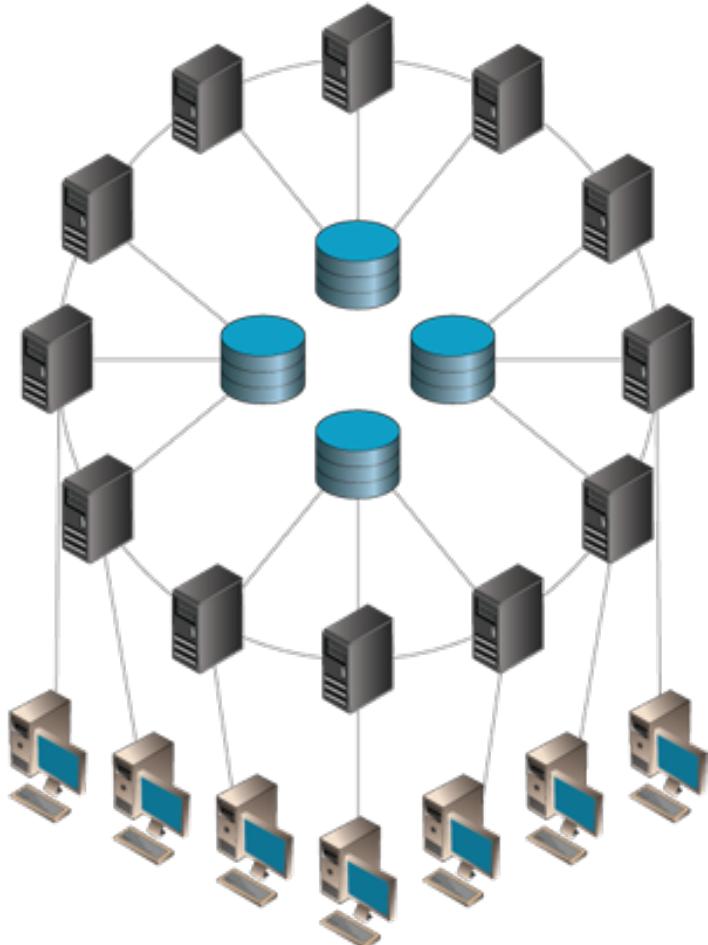
Static Analysis

a.k.a. Types

Matt's Spaceship



David's Distributed Systems



- Easy to swap in components
 - Rolling upgrade at the press of a button
- Broken views & APIs don't corrupt data
 - Temporary issues
- Rollback & hotfix are options
 - Only data layer bugs are critical

FPGA

- 6 hours to compile
- Extensive use of verification tools
- Proposed Solutions
 - Bluespec
 - CλaSH
 - Piplin
 - Chisel



Context, Requirements, and Decisions

They feed into each other

Making the Case

Section IV

I'll prove that types are best

p.s. don't do this

Types are Theorems

Programs are Proofs

a.k.a. Curry-Howard correspondence

Theorems and Proofs?



Why use types over TLA+?



Chris Meiklejohn and 1 other retweeted

adrian cockcroft @adrianco · Jun 13

#qconnewyork @swami_79 @ksshams AWS SREs learn PlusCal to work on TLA+ in a few weeks. Not an esoteric research only tool.



...

Types are integrated

Part of the language

Just press build

You'll quickly find out if the types
don't work

Types feel like less commitment

Reasonable



Insane



The Summary

If you zoned out or
ragequit in your mind

The Human Factor

Popular? Fun to use? Easy for hiring?

Beer Snobs

Haskell and PHP have their places

The Cost of Failure

Redeployment or Bankruptcy?

Remember



Chris Meiklejohn and 1 other retweeted

adrian cockcroft @adrianco · Jun 13

#qconnewyork @swami_79 @ksshams AWS SREs learn PlusCal to work on TLA+ in a few weeks. Not an esoteric research only tool.



...

There's a spectrum

Don't be dogmatic

Questions and Comments

People are Tribal

Other tribes ≠ the enemy

Tribes

