goto; conference prague

DART

an introduction

Karl Krukow,

Software Engineer, Trifork

kkr@trifork.com

software pilots
TRIFORK

About me

- PhD, 2006, Computer Science, University of Aarhus, Denmark (Theory-stuff :)
- Software Engineer at Trifork
  - Web, JavaScript, Java, Ruby, Clojure, Mobile.
  - Co-authoring a Dart book by Kresten Krab Thorup!
- Co-owner of a start-up, LessPainful, doing cloud-based automated testing for native mobile: http://www.lesspainful.com

What is Dart?

What is Dart?

- Three things
  - A new programming language, supporting structured programming for the web.
  - Set of tools supporting development with Dart.
  - Open source project

What is Dart?

- Three things
  - A new programming language, supporting structured programming for the web.
  - Set of tools supporting development with Dart.
  - Open source project
- Important to remember:
  - THIS IS A TECHNOLOGY PREVIEW (don't put it in production just yet).

Who?

Who?

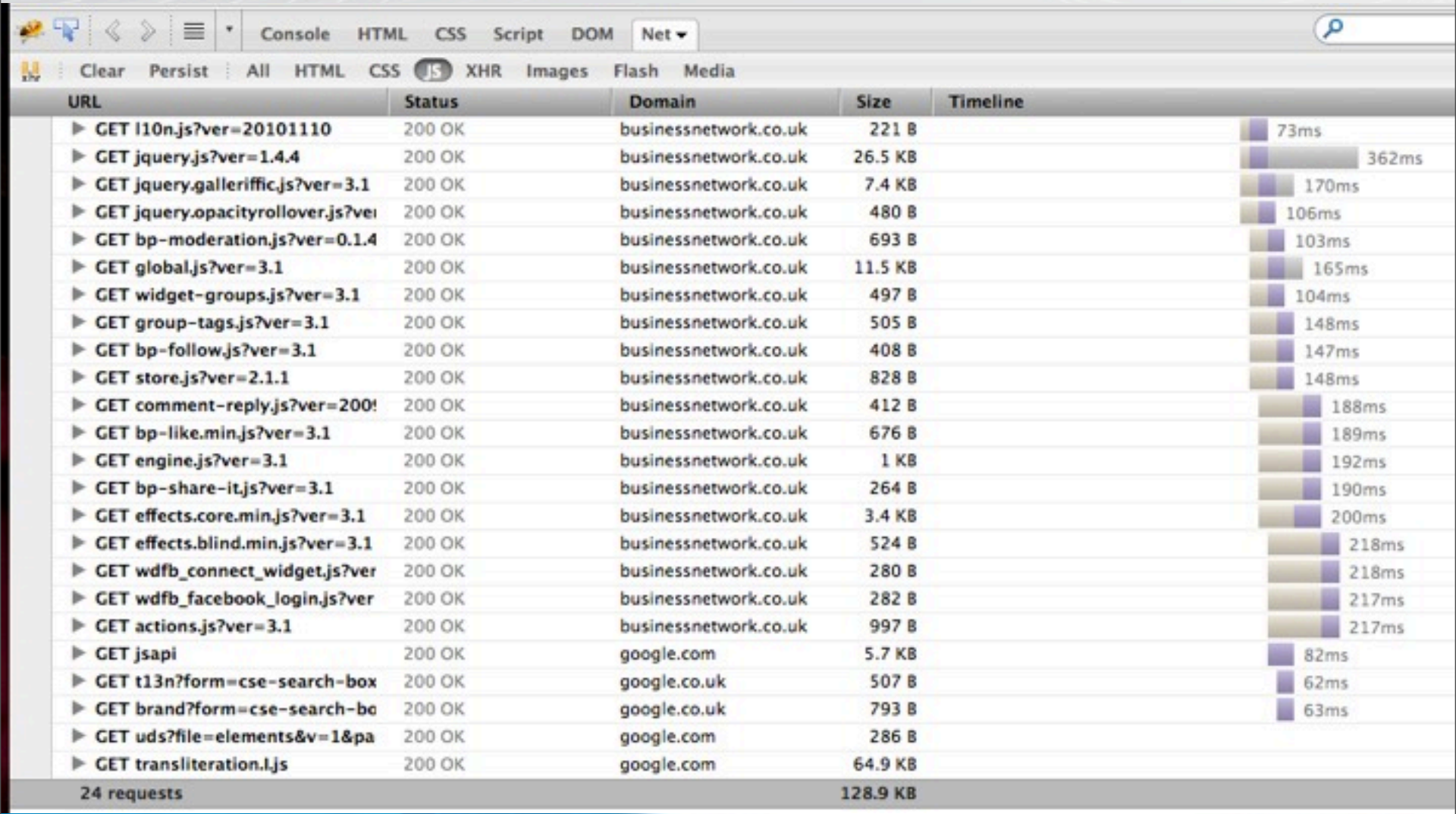
- Language and VM expert team at Google, some highlights:
  - Lars Bak (High-perf. VMs, Strongtalk, OOVM, Hotspot, V8,...)
  - Gilad Bracha (Newspeak, Strongtalk, JLS, types...)
  - Kasper Lund (CLDC HI, OOVM, V8, Spot, ...)

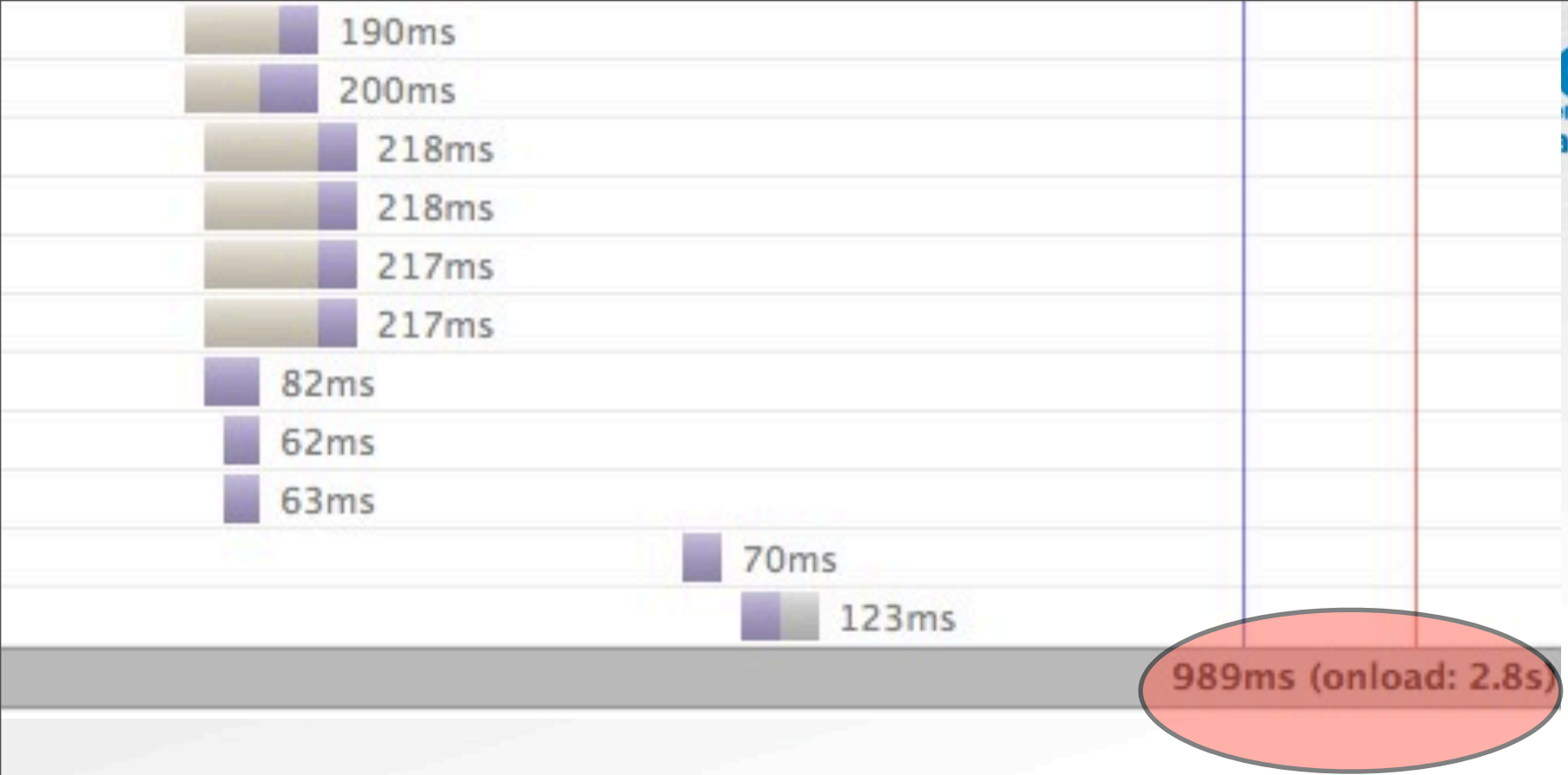
Why?

Why?

- JavaScript is focused on flexibility+ease of use
  - Hard to reason about the program structure.
  - No direct support for modularity
  - Comparatively weak tool support.
  - In the wild: unstructured, lang. mix, dependencies
- Security
  - 3rd party code runs at same priviledge as ours

In the wild: user experience


Console HTML CSS Script DOM Net
Clear Persist All HTML CSS JS XHR Images Flash Media
URL Status Domain Size Timeline
GET l10n.js?ver=20101110 200 OK businessnetwork.co.uk 221 B 73ms
GET jquery.js?ver=1.4.4 200 OK businessnetwork.co.uk 26.5 KB 362ms
GET jquery.galleriffic.js?ver=3.1 200 OK businessnetwork.co.uk 7.4 KB 170ms
GET jquery.opacityrollover.js?ver 200 OK businessnetwork.co.uk 480 B 106ms
GET bp-moderation.js?ver=0.1.4 200 OK businessnetwork.co.uk 693 B 103ms
GET global.js?ver=3.1 200 OK businessnetwork.co.uk 11.5 KB 165ms
GET widget-groups.js?ver=3.1 200 OK businessnetwork.co.uk 497 B 104ms
GET group-tags.js?ver=3.1 200 OK businessnetwork.co.uk 505 B 148ms
GET bp-follow.js?ver=3.1 200 OK businessnetwork.co.uk 408 B 147ms
GET store.js?ver=2.1.1 200 OK businessnetwork.co.uk 828 B 148ms
GET comment-reply.js?ver=200 200 OK businessnetwork.co.uk 412 B 188ms
GET bp-like.min.js?ver=3.1 200 OK businessnetwork.co.uk 676 B 189ms
GET engine.js?ver=3.1 200 OK businessnetwork.co.uk 1 KB 192ms
GET bp-share-it.js?ver=3.1 200 OK businessnetwork.co.uk 264 B 190ms
GET effects.core.min.js?ver=3.1 200 OK businessnetwork.co.uk 3.4 KB 200ms
GET effects.blind.min.js?ver=3.1 200 OK businessnetwork.co.uk 524 B 218ms
GET wdfb_connect_widget.js?ver 200 OK businessnetwork.co.uk 280 B 218ms
GET wdfb_facebook_login.js?ver 200 OK businessnetwork.co.uk 282 B 217ms
GET actions.js?ver=3.1 200 OK businessnetwork.co.uk 997 B 217ms
GET jsapi 200 OK google.com 5.7 KB 82ms
GET t13n?form=cse-search-box 200 OK google.co.uk 507 B 62ms
GET brand?form=cse-search-bo 200 OK google.co.uk 793 B 63ms
GET uds?file=elements&v=1&pa 200 OK google.com 286 B
GET transliteration.I.js 200 OK google.com 64.9 KB
24 requests 128.9 KB


190ms
200ms
218ms
218ms
217ms
217ms
82ms
62ms
63ms
70ms
123ms
989ms (onload: 2.8s)

Start-up performance is often really bad.

Potential advantages of Dart

Potential advantages of Dart

- Tool support
- Fairly “low ceremony” with (optional) types
- Unified language
- Concurrency without “lock-and-pray”.
- Security (sandbox + capabilities)
- Performance
- Simplified library interoperability (“imports”)

goto; conference prague

The Dart Programming Language

Main language features

Main language features

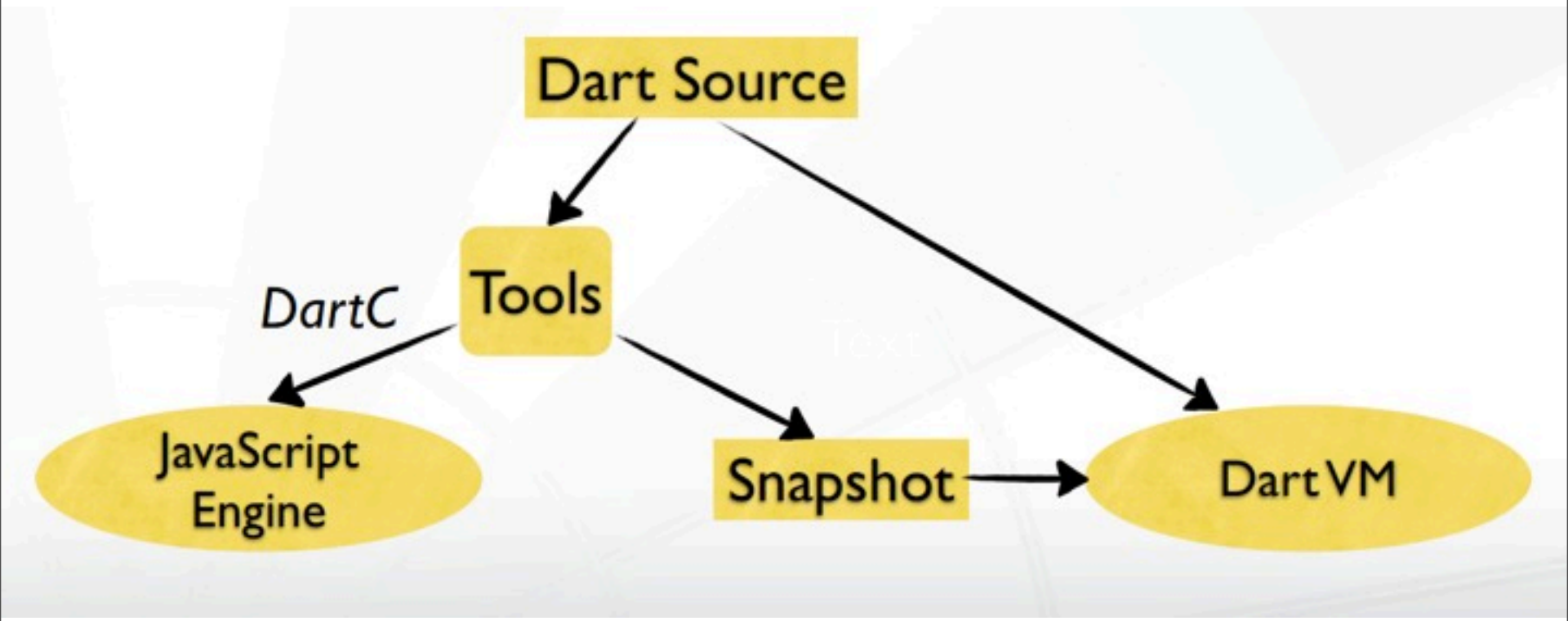
- Pure object-oriented dynamic programming language. Familiar syntax.
  - Class-based single inheritance with interfaces.
  - Implicit interfaces, factories.

Main language features

- Pure object-oriented dynamic programming language. Familiar syntax.
  - Class-based single inheritance with interfaces.
  - Implicit interfaces, factories.
- Optional static typing

Main language features

- Pure object-oriented dynamic programming language. Familiar syntax.
  - Class-based single inheritance with interfaces.
  - Implicit interfaces, factories.
- Optional static typing
- Isolates: message-passing model for concurrency
  - Isolates are single threaded.
  - basis for a capability-based security model



Dart on server: typically on dartvm

Dart on client: dartvm in browser, JS in browser

goto; conference prague

Let's see it!

Classes, interfaces, closures.
Optional types.

More on Dart's optional types

More on Dart's optional types

- Annotations don't affect runtime semantics!

More on Dart's optional types

- Annotations don't affect runtime semantics!
- Controversial: Type system is actually 'unsound', e.g., co-variant generics.
  - "Ergonomically incorrect" - G. Bracha

More on Dart's optional types

- Annotations don't affect runtime semantics!
- Controversial: Type system is actually 'unsound', e.g., co-variant generics.
  - "Ergonomically incorrect" - G. Bracha
- Reified types: type args to constructors, interfaces.

More on Dart's optional types

- Annotations don't affect runtime semantics!
- Controversial: Type system is actually 'unsound', e.g., co-variant generics.
  - "Ergonomically incorrect" - G. Bracha
- Reified types: type args to constructors, interfaces.
- "Checked mode": runtime checks of annotations.

Ergonomically incorrect

Ergonomically incorrect

```
public static <T extends Object & Comparable<? super T>>
                    T max(Collection<? extends T> coll) {
    ...
}
```

Ergonomically incorrect

```
public static <T extends Object & Comparable<? super T>>
                    T max(Collection<? extends T> coll) {
    ...
}
```

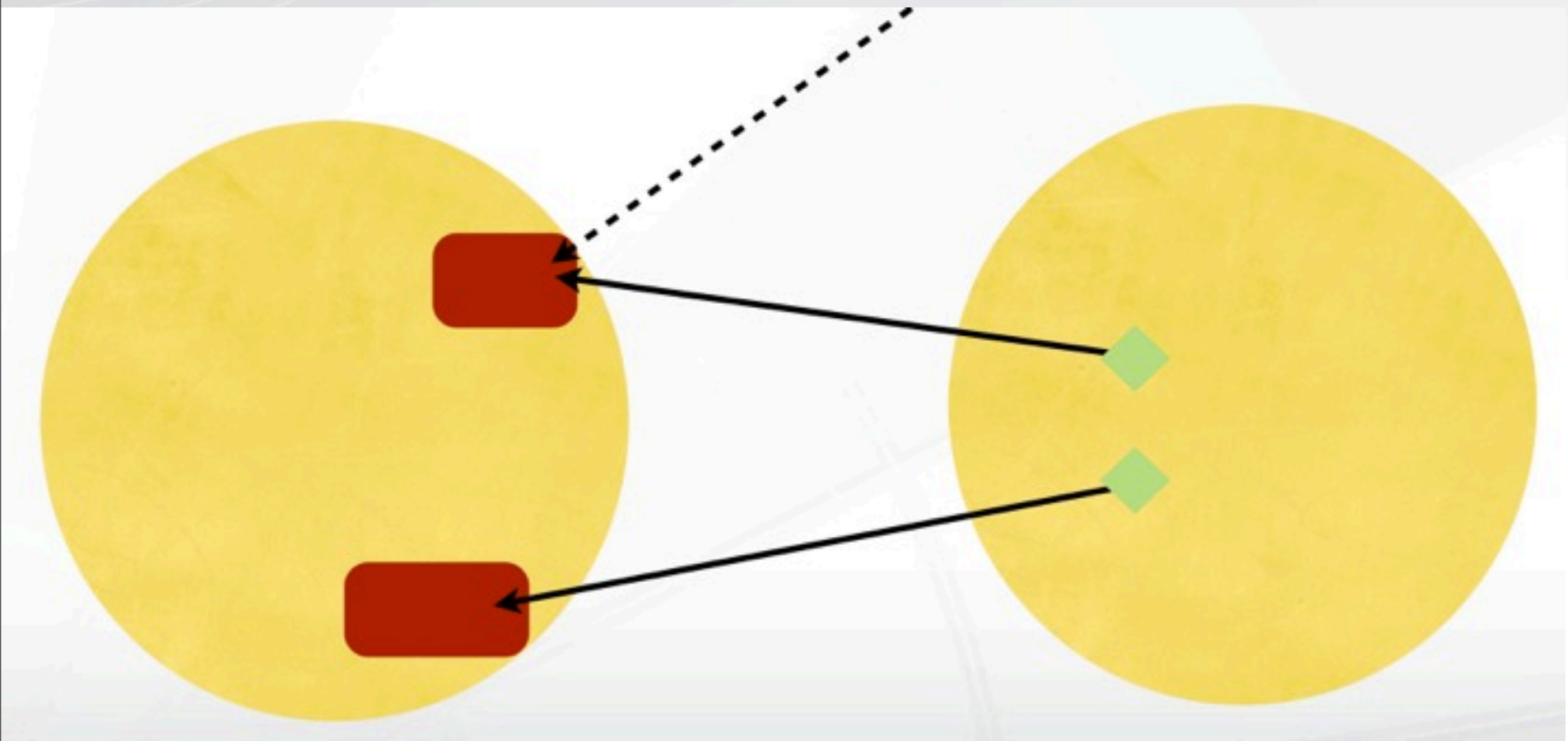
vs

```
Comparable max(List<Comparable> list) {
  ...
}
```

Isolates

- Lightweight units of execution.
- Runs in their own address space like processes.
- All communication takes place via message passing
- Each isolate is “sequential”/single threaded
  - Dart supports concurrent execution by spawning multiple isolates.
- Only message queues shared, no locks needed!

Isolates



More on ports

- Receive ports accept and enqueue incoming messages
  - An isolate is born with a receive port.
  - Can also be created on demand.
- A send port allows sending to a certain receive port
  - It is an unforgeable, transferrable capability
  - Basis for sandboxing untrusted code.

goto; conference prague

Let's see it!

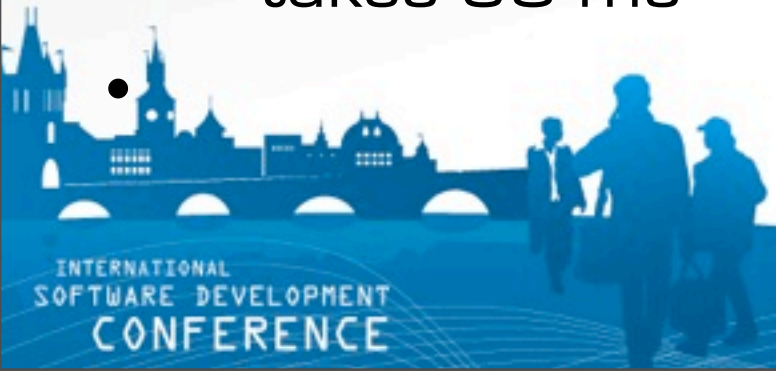
Isolates

Performance

- Now, DartVM isn't generally faster than V8.
- But Dart is likely to go faster than V8 will
  - due to simplified semantics of Dart compared to JavaScript.

Snapshots

- Serialized heap after loading the application
- Startup can be more than 10x faster
- Example:
  - Loading 54173 lines of Dart code takes 640 ms
  - Loading same application from a snapshot takes 60 ms
-



Open source project

- The Dart web site: http://dartlang.org
  - Dart language specification
  - Dart language tutorial + articles
- The Dart project: http://dart.googlecode.com
- Libraries and code samples
- Dart virtual machine+ JavaScript compiler(s)
- http://try-dart-lang.appspot.com

DART IS NOT DONE

- Give feedback by joining the discussion!
  - Some of the features aren't implemented yet...
  - Rest arguments and enums?
  - Reflection support?
  - Isolates are kind-of low-level...

DART

goto; conference prague

Watch out for the upcoming Dart book!

Thanks.
Questions?

Surprisingly Unsurprising?

Surprisingly Unsurprising?

- Why isn't Dart more like:
  - Clojure, Erlang, Haskell, Scala, <exotic language X>

Surprisingly Unsurprising?

- Why isn't Dart more like:
  - Clojure, Erlang, Haskell, Scala, <exotic language X>
- Deliberately designed to be familar:
  - developed with massive-adoption as primary goal.

Surprisingly Unsurprising?

- Why isn't Dart more like:
  - Clojure, Erlang, Haskell, Scala, <exotic language X>
- Deliberately designed to be familar:
  - developed with massive-adoption as primary goal.
- Also, focus on enabling high performance and powerful tooling.

  + Must compile to efficient JavaScript.