Dart: a modern web language

Nicolas Geoffray
Google

Who am I?

Nicolas Geoffray, software engineer at Google

Projects

- **VVM** - Highly dynamic runtime environment
- **VMKit** - Framework for writing VMs
- **I-JVM** - Better dependability in OSGi
- **Dart** - Structured programming for the web

DART

Motivation

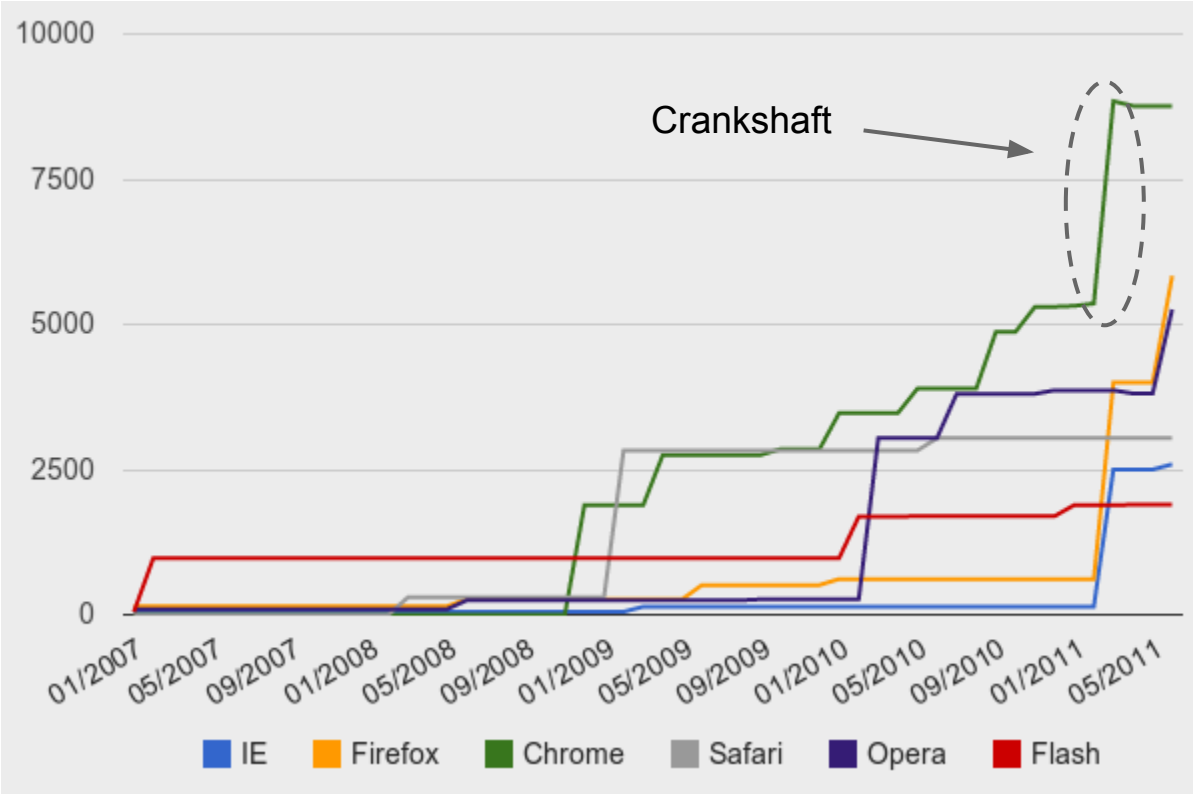
Improve web development

The web is already pretty awesome

- It is easy to develop small applications
  - Code runs everywhere (phones, desktops)
  - No installation of applications
  - Deployment is almost trivial
- JavaScript is very flexible and supports incremental development

DART

The rise of JavaScript



Credit: http://iq12.com/blog/

DART

Why is the web hard to program for?

- Writing large well-performing applications is hard
- Hard to reason about the program structure
- Startup performance is often really bad
- Difficult to document intent (lack of types)
- No support for modules, packages, or libraries

DART

Make it easier

- We want to improve the web platform
  - Better support for programming in the large
  - Faster application startup (especially on mobile)
  - More predictable and better runtime performance
  - JavaScript is a powerful tool but it has sharp edges
- Keep up the innovation momentum
  - The web is evolving at a fantastic pace!
  - The developer tools have to keep up

DART

JavaScript is full of ... surprises

- Lots and lots of implicit type conversions
- Most operations produce weird results when passed wrong or uninitialized values instead of failing in a recognizable way

Keep on truckin'

DART

No argument type checking

```
var x = 499;
x + null;
x + [];
x + undefined;
x - {};
```

No argument type checking

```
var x = 499;
x + null;  // => 499
x + [];
x + undefined;
x - {};
```

No argument type checking

```
var x = 499;
x + null;  // => 499
x + [];  // => "499"
x + undefined;
x - {};
```

No argument type checking

```
var x = 499;
x + null;  // => 499
x + [];  // => "499"
x + undefined;  // => NaN
x - {};
```

No argument type checking

```
var x = 499;
x + null;  // => 499
x + [];  // => "499"
x + undefined;  // => NaN
x - {};  // => NaN
```

No array bounds checking

```
var array = new Array(32);
...
array[32];
array[-1];
array[.1];
array[null];
array[array];
```

No array bounds checking

```
var array = new Array(32);
...
array[32];  // => undefined
array[-1];  // => undefined
array[.1];  // => undefined
array[null];  // => undefined
array[array];  // => undefined
```

No array bounds checking

```
var array = new Array(32);
...
array[32];  // => void 0
array[-1];  // => void 0
array[.1];  // => void 0
array[null];  // => void 0
array[array];  // => void 0
```

No spell checking?

```
var request = new XMLHttpRequest();
...
request.onreadystatechange = function() {
  if (request.readystate == 4) {
    console.log('Request done!');
  }
};
```

No spell checking?

```
var request = new XMLHttpRequest();
...
request.onreadystatechange = function() {
  if (request.readyState == 4) {
    console.log('Request done!');
  }
};
```

JavaScript has improved but ...

- JavaScript has fundamental issues at the language level that impact productivity
- Performance has improved but mostly for a pretty static subset of JavaScript
- It remains very time consuming to build and maintain large web apps

The story of Dart

- A few years ago Lars Bak and Kasper Lund prototyped Spot
  - A new simple programming language for the web
  - Based on their experiences from JavaScript/V8
- Spot was the prelude for the Dart project

What is Dart?

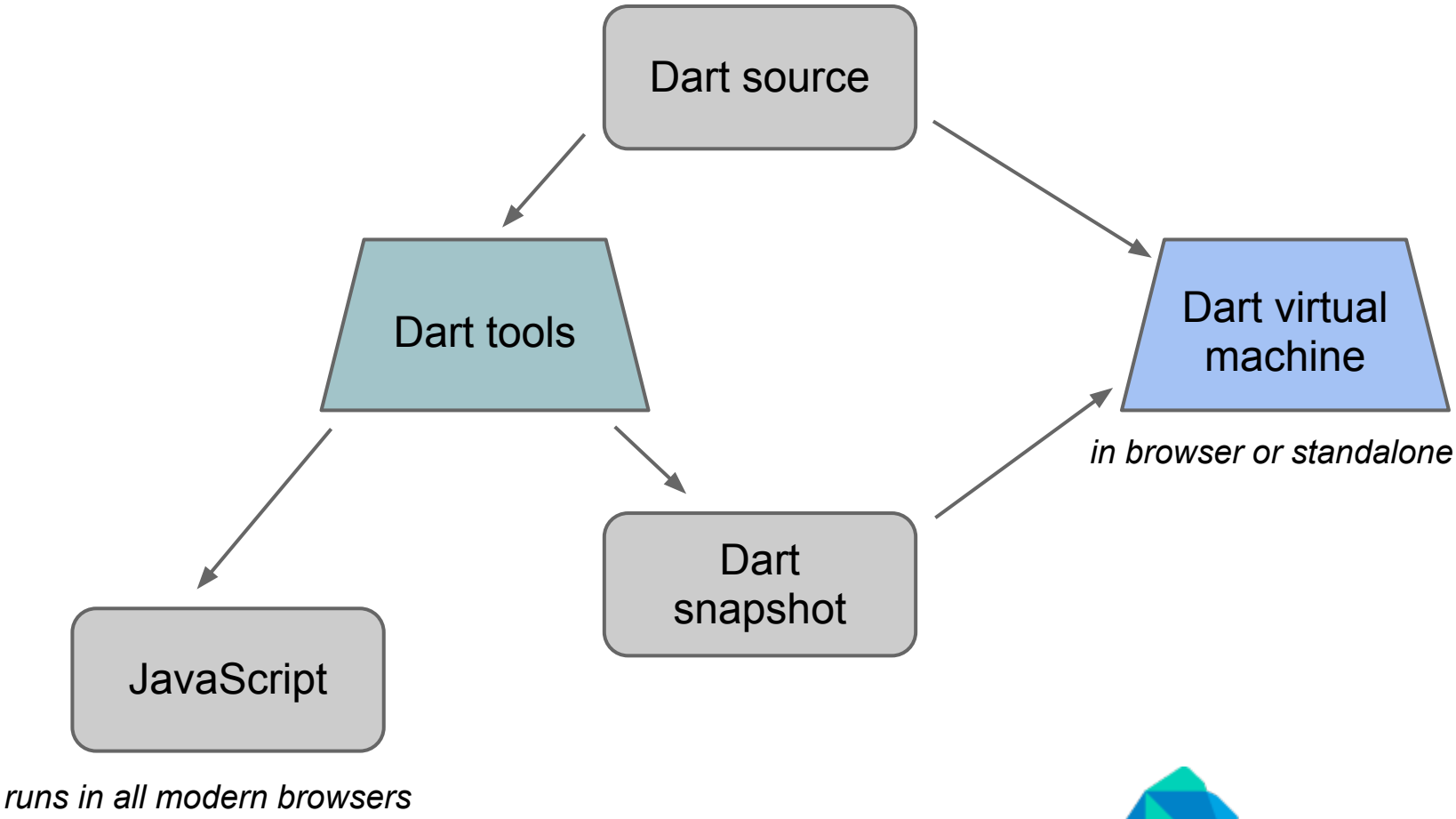
- Unsurprising object-oriented programming language
- Class-based single inheritance with interfaces
- Familiar syntax with proper lexical scoping
- Single-threaded with isolate-based concurrency
- Optional static types

And more!

Dart comes with a lot of developer tools:

- DartEditor: Eclipse based Dart editor
- Dartium: Chromium with embedded Dart VM
- dart2js: Dart-to-JavaScript compiler
- Libraries: io, crypto, i18n, ...

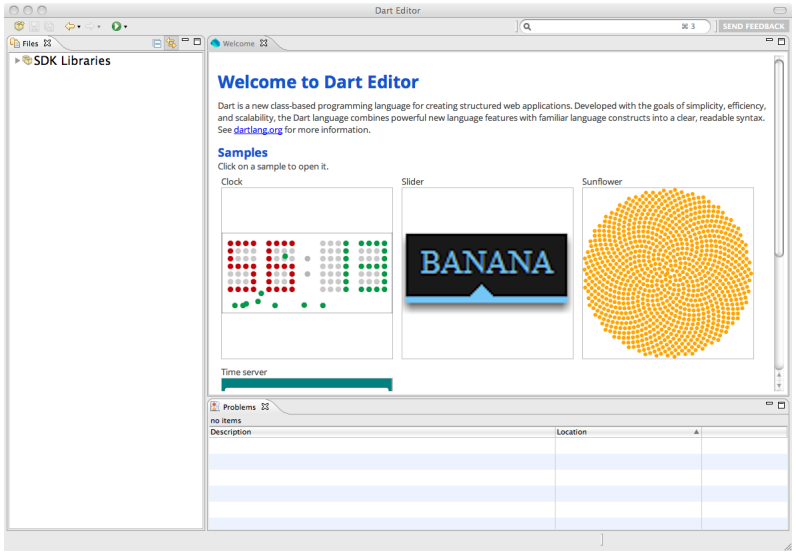
Deployment and execution



DART

Let's see it in action

- Let's write simple applications with the Eclipse-based Dart Editor



Conventional type checking

- Tries to prove that your program obeys the type system
- Considers it a fatal error if no proof can be constructed
- In Dart, you are innocent until proven guilty...

```
List<Apple> apples = tree.pickApples();
printFruits(apples);

void printFruits(List<Fruit> fruits) {
  for (Fruit each in fruits) print(each);
}
```

DART

Optional static types

- Static types convey the intent of the programmer
- Checkable documentation for code and interfaces
- Avoids awkward variable naming or comment schemes
- Type annotations have no effect on runtime semantics

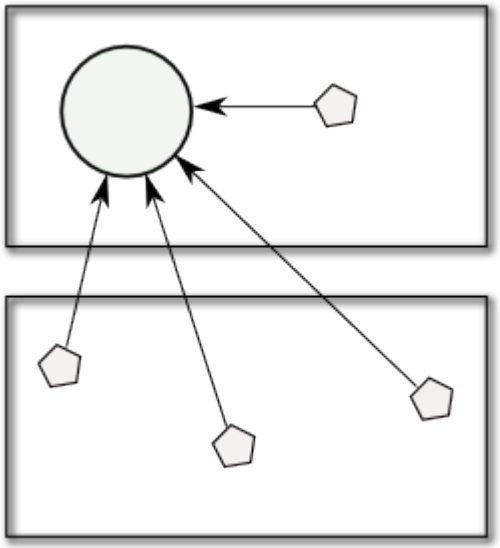
Isolates

Isolates are lightweight units of execution:

- Run in their own address space like processes
- Nothing is shared - nothing needs synchronization
- All communication takes place via messaging passing
- Supports concurrent execution

Communication

- ReceivePorts:
  - enqueues incoming messages
  - can not leave their isolate
  - can be created on demand



- SendPorts:
  - created by a ReceivePort
  - dispatches messages to its ReceivePort
  - can be transferred (across Isolate boundaries)
  - Unforgeable, transferable capability

DART

Dart virtual machine

- Dart has been designed for performance
  - Simplicity gives more performance headroom
  - Enforced structure leads to better predictability
  - Virtual machine performs better than V8 at launch
- Works embedded in browser or standalone
  - Experimental Dart-enabled build of Chromium
  - SDK includes preliminary server-side libraries

```
$ dart hello.dart
```

Dart-to-JavaScript

- Compiler is implemented in Dart
  - Generates JavaScript that runs in modern browsers
  - SSA based optimizations
  - Uses *tree shaking* to cut down on code size

```
$ dart2js --out=hello.js hello.dart
```

Flavour of generated JavaScript

```
class Point {
  var x, y;
  Point(this.x, this.y);
  toString() => "($x,$y)";
}
```

```
Isolate.$defineClass("Point", "Object", ["x", "y"], {
  toString$0: function() {
    return '(' + $.toString(this.x) + ',' +
                 $.toString(this.y) + ')';
  }
});
```

DART

Performance

Goal chart	Scores			Relative to v8	
Benchmark	v8	dart	dart2js	dart	dart2js
DeltaBlue	279.72	368.50	190.31	131.74%	68.04%
Richards	400.30	566.22	281.36	141.45%	70.29%
NBody	15944.00	17513.50	10876.00	109.84%	68.21%
BinaryTrees	9.01	9.35	8.24	103.79%	91.47%
Mandelbrot	169.33	167.92	138.29	99.16%	81.67%
Fannkuch	3465.00	4325.50	3142.00	124.83%	90.68%
Meteor	6.69	5.60	2.19	83.75%	32.81%
BubbleSort	25237.50	26449.00	18222.00	104.80%	72.20%
Fibonacci	9198.50	13534.00	9405.50	147.13%	102.25%
Loop	34889.50	35319.00	35469.00	101.23%	101.66%
Permute	11082.00	16535.00	7519.50	149.21%	67.85%
Queens	117959.99	181779.51	98879.00	154.10%	83.82%
QuickSort	17107.50	15312.50	9403.50	89.51%	54.97%
Recurse	14019.50	20194.00	14424.00	144.04%	102.89%
Sieve	102290.50	114639.50	102462.50	112.07%	100.17%
Sum	74423.50	59832.00	75394.00	80.39%	101.30%
Tak	3064.00	4763.50	2490.00	155.47%	81.27%
Takl	8910.00	16699.50	8431.00	187.42%	94.62%
Towers	4919.50	5611.00	3107.00	114.06%	63.16%
TreeSort	7041.00	7933.00	5427.00	112.67%	77.08%
Geo. mean	4009.77	4785.62	3120.74	119.35%	77.83%

DART

Open source

- Dart is available under a BSD license
- Developed in the open (code reviews, build bots, etc.)

Online resources

- **Primary site** - http://www.dartlang.org/
- **Code** - http://dart.googlecode.com/
- **Libraries** - http://api.dartlang.org/
- **Specification** - http://www.dartlang.org/docs/spec/

Summary

- Dart is an unsurprising, object-oriented language that is instantly familiar to most
- Dart allows you to write code that tools and programmers can reason about
- Dart applications runs in all modern browsers through translation to JavaScript

Dart was designed with performance in mind.

Dart allows rapid prototyping and structured development.

Thank you!

Dart is open source and instantly familiar to lots of programmers.

Dart runs everywhere JavaScript does.