



DART

Google™

Anders Sandholm



DART: A STRUCTURED WEB PROGRAMMING LANGUAGE

- *Public preview Oct 10*
- *A programming language*
- *Programming tools*
- *Open source project*



SPEAKER INTRODUCTION

Anders Sandholm



V8
Devtools
Dart



Google™

AGENDA

- *Motivation*
- *Language*
- *Samples*
- *Tools*
- *Open source project*



Google™

CURRENT WEB: THE GOOD PARTS

- *Developing small applications is easy*
 - *Platform independence*
 - *No installation of applications*
 - *Supports incremental development*
 - *... and it is everywhere*



Google™

CURRENT WEB: THE BAD PARTS

- *Developing large scale applications is hard*
 - *Hard to find the program structure*
 - *Lack of static types*
 - *No support for libraries*
 - *Tools support is weak*
 - *Startup performance is bad*



WHAT ARE THE GOALS FOR DART?

- *Real support for programming in the large*
- *Ultra-fast startup*
- *Predictable performance*
- *Incremental execution*
- *Backwards compatibility*



Google™

THE DART PROGRAMMING LANGUAGE

A simple and unsurprising OO programming language

- *Class-based single inheritance*
- *Interfaces with default implementation*
- *Optional static types*
- *Real lexical scoping*
- *Single-threaded*



Google™

LET'S TRY SOME DART CODE

- *Fun with classes, closures, and optional types*
- *Easy to experiment with at try.dartlang.org*



Google™

A DIFFERENT TYPE-CHECKER

- *A conventional type-checker is a simplified theorem prover*
 - *Tries to prove programs obey type system*
 - *If it can't construct a proof - program is considered invalid*
"Guilty until proven innocent"
- *In Dart, you are **innocent until proven guilty***



Google™

OPTIONAL STATIC TYPES

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



TYPES ON THE DARTBOARD

- *Let's explore a few illustrative examples*



Google™

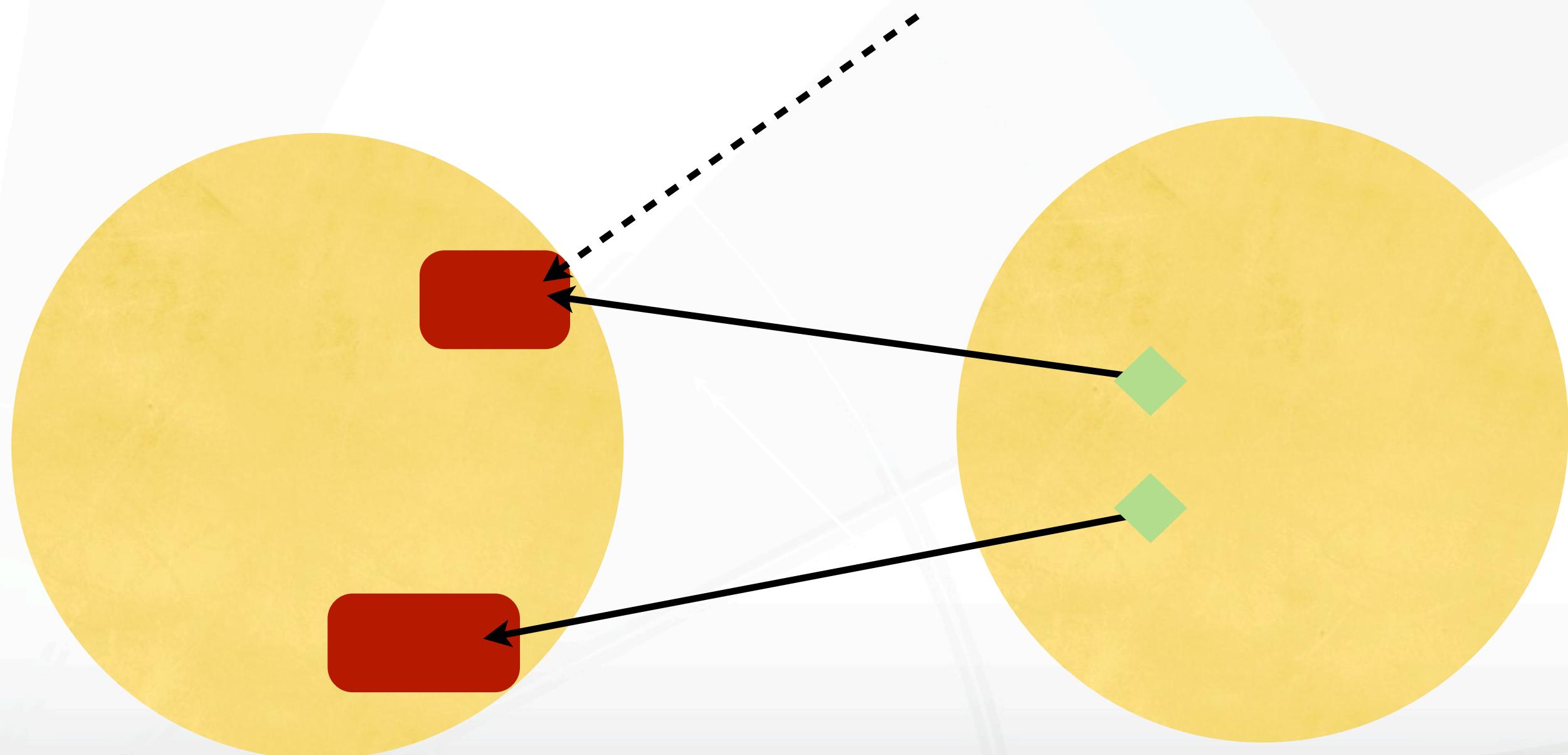
ISOLATES

- *Lightweight units of execution*
 - *Runs in their own address space like processes*
 - *Nothing is shared - nothing needs synchronization*
 - *All communication takes place via message passing*
- *Supports concurrent execution*



Google™

SENDING AND RECEIVING MESSAGES



DART

Google™

PORTS

- *Receive ports accept and enqueue incoming messages*
 - *Live inside a specific isolate*
 - *Can be created on demand*
- *A send port allows sending to a certain receive port*
 - *It is an unforgeable, transferrable capability*



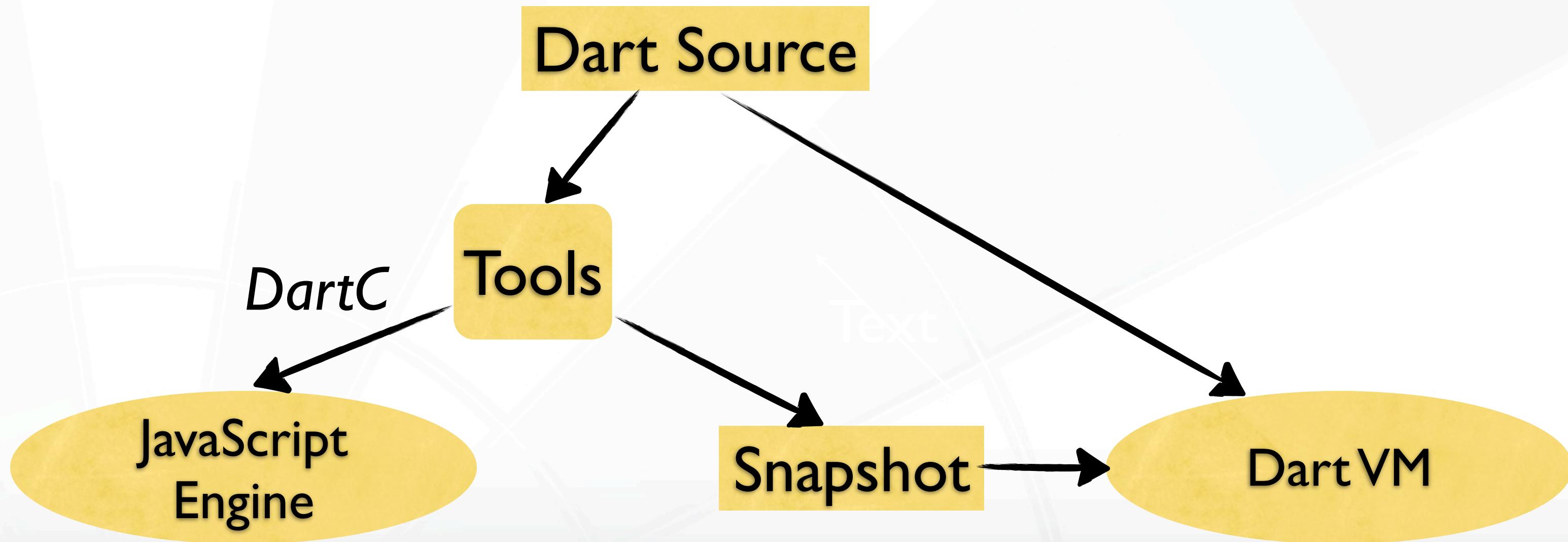
Google™

BACKWARDS COMPATIBILITY

- *Breaking the web is not an option*
- *Translation strategy to JavaScript is crucial*
- *Performance when translated must be comparable*



DART EXECUTION



Google™

BACKWARDS COMPATIBILITY IMPACT

- *No non-local-return*
- *Threading not possible*
- *Number hierarchy*

num
integer
double



Google™

PROGRAMMING IN THE LARGE

- *Flexible development modes*
 - *Static type checking*
 - *Runtime type validation*
- *Declared application and libraries*
- *Isolates, actor-like concurrency model*



Google™

DART PERFORMANCE

Relative performance compared to JavaScript on V8

Benchmark	VM	Dart->JS Compiler
Mandelbrot	18.1%	101.0%
DeltaBlue	60.5%	85.0%
Richards	49.9%	79.9%
NBody	37.5%	83.2%
BinaryTrees	70.3%	99.9%
Fannkuch	58.4%	78.9%
Meteor	48.2%	99.4%

Details:

- V8 revision 3.5.5.
- Dart revision 1331



Google™

WEB APPLICATION IN DART

- *Newsreader completely written in Dart*
- *App code: 3210 LOC*
- *UI library code: 13200 LOC*
- *Animation yields 30 fps*
- *Code is part of the open source project*



Google™

DART EDITOR

- *Editor for constructing and browsing Dart applications*
- *Lightweight editor based on Eclipse components*
- *Code is part of the open source project*



Google™

DART OPEN SOURCE PROJECT

- *The Dart web site: <http://dartlang.org>*
 - *Dart language specification*
 - *Dart language tutorial*
- *The Dart project: <http://dart.googlecode.com>*
 - *Libraries and code samples*
 - *Dart virtual machine*
 - *Dart to JavaScript compiler*



CREDIT

- *Object model inspired by **Smalltalk***
- *Compilation strategy inspired by **Self***
- *Optional types inspired by **Strongtalk***
- *Isolates design inspired by **Erlang***
- *Syntax inspired by **JavaScript & C***



Google™

DART SUMMARY

- *Technology preview*
- *A programming language for the web*
 - *Simple and clean execution model*
 - *Designed with compatibility in mind*
- *Open source project*



Google™



DART

Q & A

Google™