Web Components

Reactive Architecture for the Front End

Steven Skelton

Reactive Programming Toronto
December 3, 2014

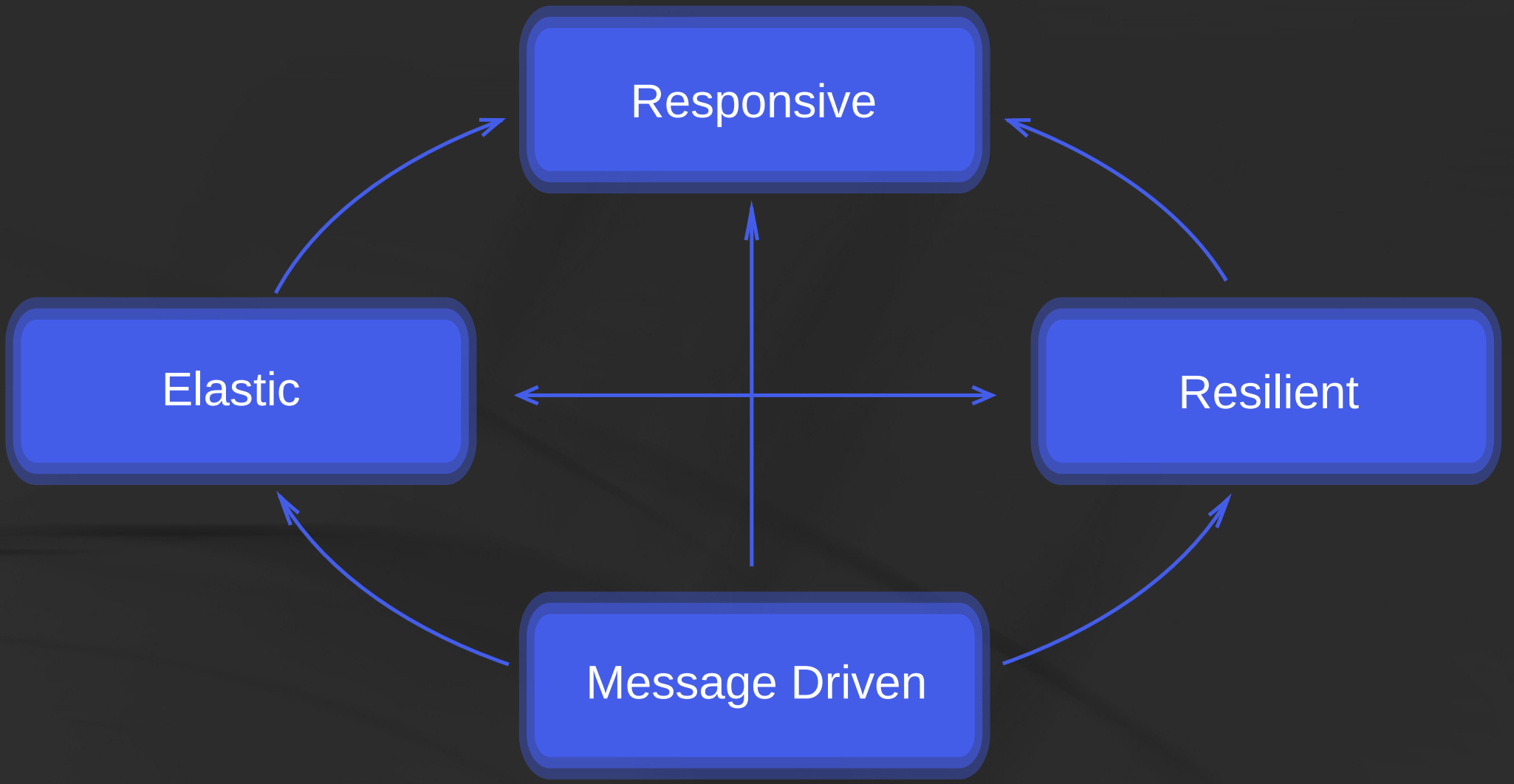
Reactive Manifesto

Is a pattern for building software capable of handling today's application requirements:

- Highly responsive to user,
- Large amounts of data,
- Grow and adapt to change

Sounds like it could also apply to the front-end.....

Reactive System Strategy



Reactive Analogue

Aggregate → Single

Server Cluster	Web Page
• Responsive	• Responsive
• Elastic	• ***Consistently Fast UI?***
• Resilient	• ***Less bugs in SDLC?***
• Message Driven	• ***Rethink Event Bindings?***

“Elastic” Browser Performance

Today websites handle **more data,**
with **rich interactions,** all within a **single page**.

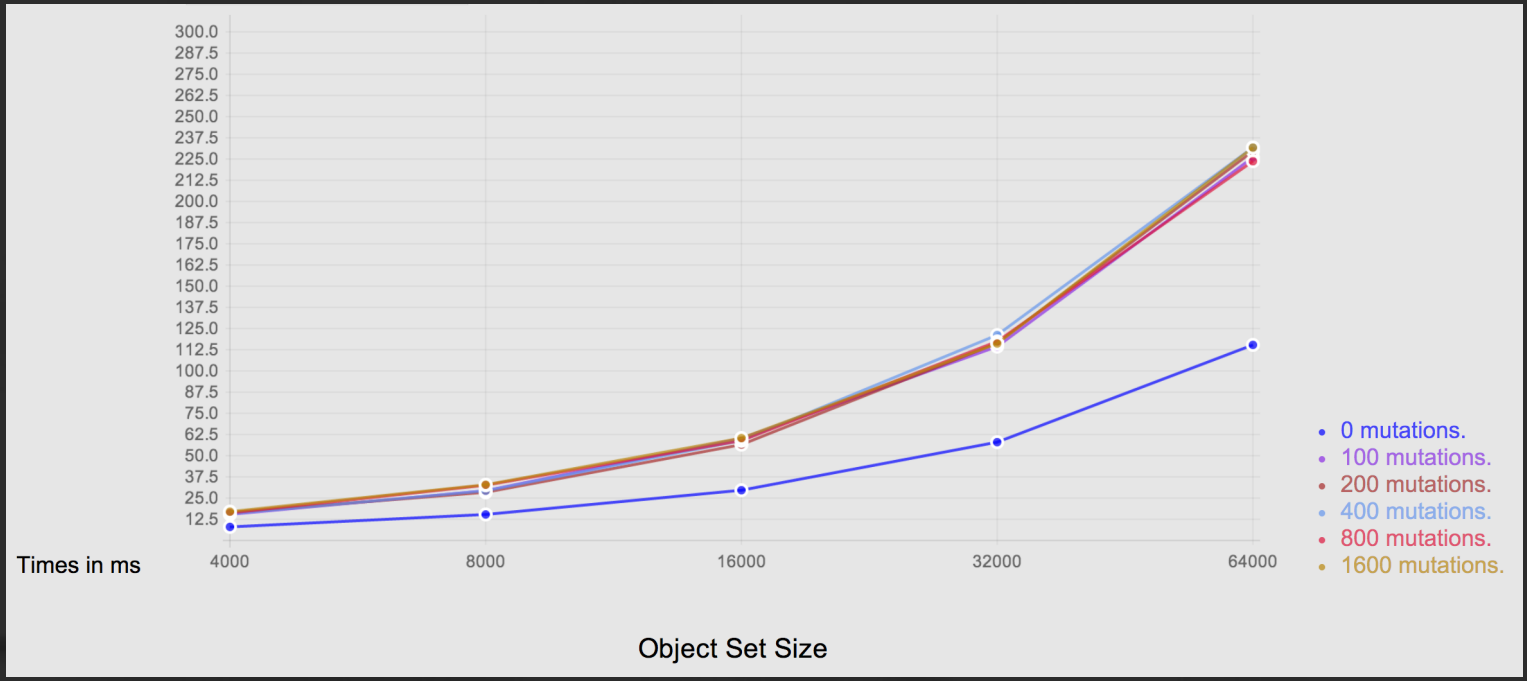
Each page requires:

- more JS libraries,
- larger downloads,
- data-binding,
- dynamic DOM.

Leading to too-many, bizarre, frameworks.

Exponential Problem

Today's frameworks bend around browser limitations



AngularJS Dirty Checking

They work...
for small data sets or a limited number of instances.

New JavaScript Features that Scale

Native Browser support

(No download, No JavaScript engine)

- DOM Mutation Observers
- JavaScript Object observers
- <template>

Multi-threading

- Web workers: Separate computations and UI
- **`async`** loading and parsing

“Resilient” Webpages

Fewer errors by simplifying code

- HTML is the final product,
- Impedance mismatch of complex frameworks.

Increase composibility, reusability of components

- Dev teams work more efficiently,
- Less work required,
- More cohesive design.

Slimmer Frameworks, Expand HTML

Native support for Custom HTML tags

- Familiar, reusable components,
- No external dependencies,
- Built using HTML DOM in a <template>.

Replaces frameworks' abstractions:

- None are framework agnostic,
- Rely on string, JSON or non-HTML templates.

Shadow DOM

Enforces proper scoping,

- Segregate HTML along bounded contexts,
  - Encapsulation of internals,
  - Guard against external CSS, Id naming conflicts,
  - Restrict interactions to messages.

Ensures Loose-coupling, Reusability

Can be pierced unlike an `iframe` when necessary

Web Components

Native Browser Support

(polyfilled if not yet supported)

- Custom HTML Elements
- **`<template>`**
- Shadow DOM
- **`Object.observe`**
- HTML Imports

Polymer

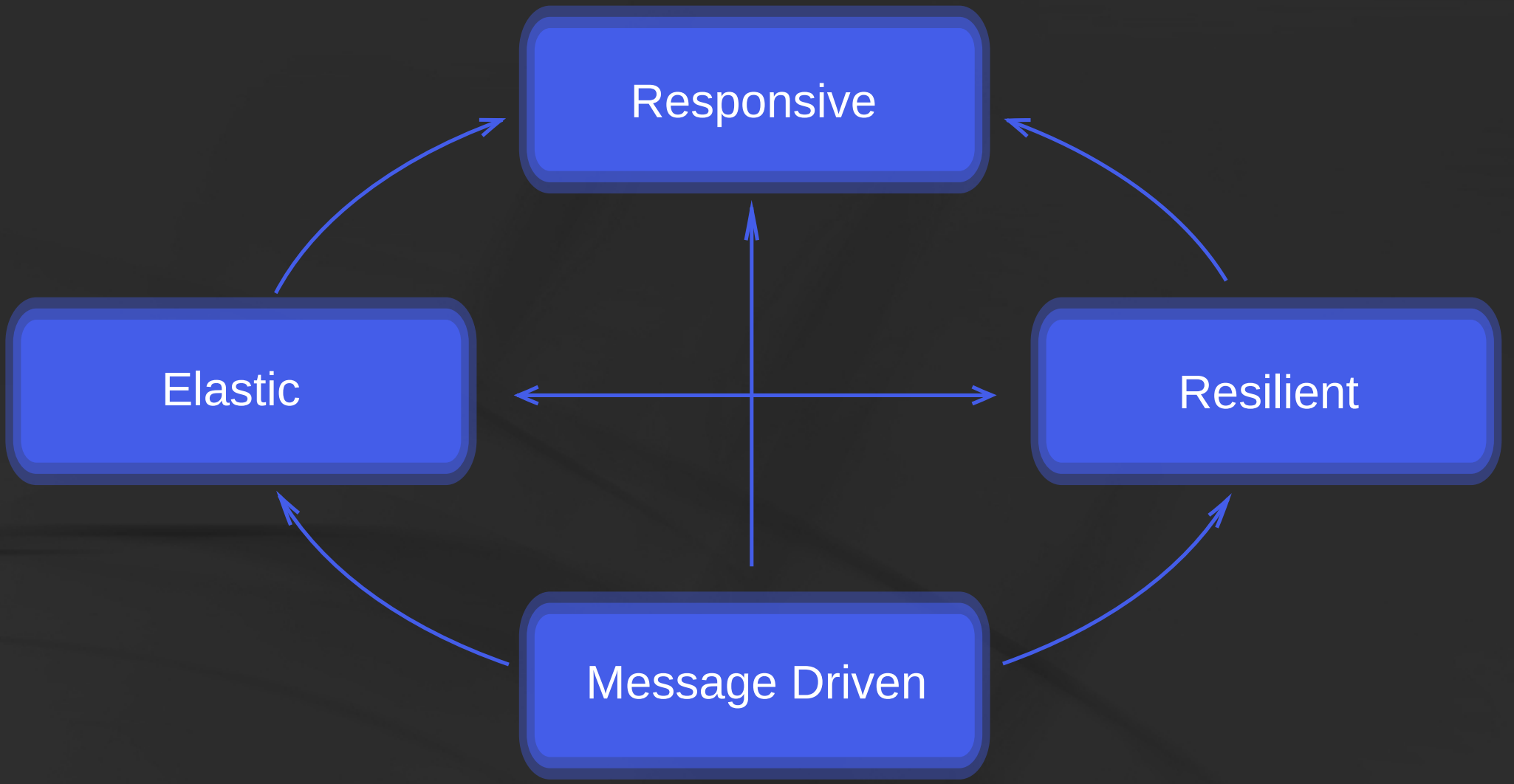
(active development at Google)

Native browser implementations are low-level and un-opinionated, **Polymer** is the first library to put it all together.

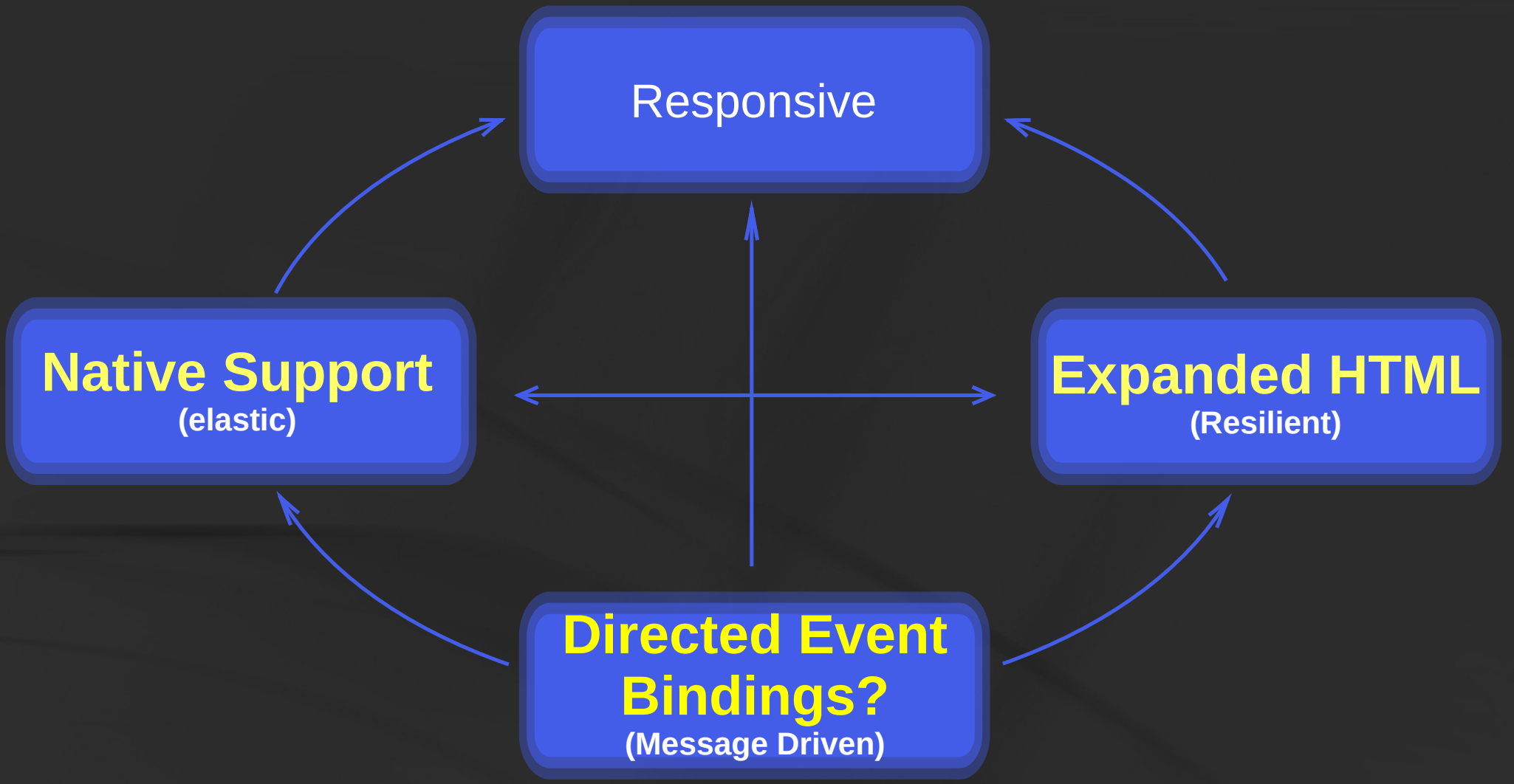
Still pre-release, version 1.0 scheduled for Q2

Already used in production at:
Salesforce, News Corp, GitHub, YouTube, Google

BE Reactive System Strategy



FE Reactive System Strategy



Message vs. Event System

Message-driven: focus on **recipients**,

Event-driven: focus on **sources**.

- Difference very small in a FE context
- Ideal system:
  - Message-oriented middleware,
  - Event-driven components,
  - Non-blocking execution.

Message Delivery via 2-Way Bindings

```
<h1>Welcome back {{account.name}}!</h1>

<song-search
    genres="{{account.genres}}"
    results="{{playlist}}">
</song-search>

<media-player playlist="{{playlist}}">
</media-player>

<div class="footer">
    <account-profile account="{{account}}">
    </account-profile>
</div>
```

observe

- Data references act as middleware,
- Location transparency,
- Components do not require knowledge of external events, maintaining loose coupling.

Polymer **expressions** and **filters** can be used to transform and combine data outside of components:

```
{{ a + b | base16 }}
{{ myNumbers | gt(3) | sort }}
```

Object.observe(), Array.observe()

Events when:

- the value of a data property is updated,
- any property is added,
- any property is deleted,
- any property is reconfigured.

No expensive dirty-checking, no special methods.

Plain Old JavaScript Objects.

High level `observe` libraries

observe.js

- PathObserver, ArrayObserver, ObjectObserver, CompoundObserver, ObserverTransform

watchtower.js

- Angular 2.0's change detection
- Undocumented, more complex

Both can Polyfill with dirty-checking

Reactive Analogue

Aggregate → Single

Server Cluster

- Responsive
- Elastic
- Resilient
- Message Driven

Web Page

- Responsive
- ***Native Support, Multi-threaded***
- ***Loose Coupling, Encapsulation***
- ***Event Driven Messages***

Strategies for a Reactive FE

Aggregate → Single

Server Cluster

- Responsive
- Elastic
- Resilient
- Message Driven

Web Page

- Responsive
- ***Native Support, Web Workers***
- ***Shadow DOM, Custom Elements***
- ***Object.observe***

Web Component Resources

- polymer-project.org
- HTML5Rocks.com
- customelements.io
- component.kitchen
- W3C Web Components wiki
- YouTube:
    - Chrome Dev Summit, Google I/O, JSConf

Thanks for Listening!

Questions ?

Blog: http://stevenskelton.ca

https://github.com/stevenrskelton

- Various Polymer web components