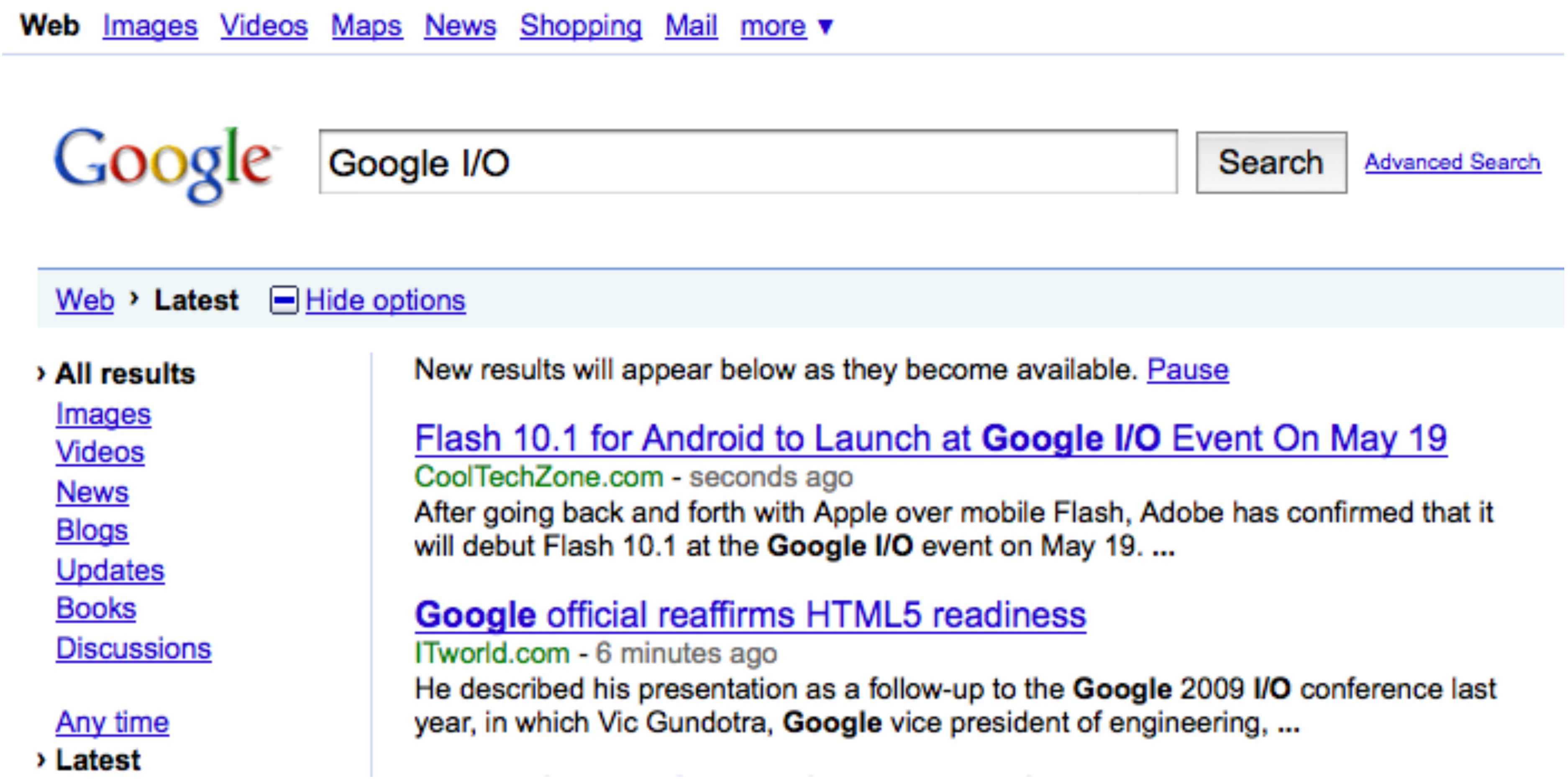
Building real-time web apps with App Engine and the Feed API

Brett Bavar
Moishe Lettvin
May 20, 2010

View live notes and ask questions about this session on Google Wave

http://bit.ly/dc6QxD

The real-time web



Google I/O

New results will appear below as they become available. Pause

Flash 10.1 for Android to Launch at Google I/O Event On May 19

CoolTechZone.com - seconds ago

After going back and forth with Apple over mobile Flash, Adobe has confirmed that it will debut Flash 10.1 at the **Google I/O** event on May 19. **...**

Google official reaffirms HTML5 readiness

ITworld.com - 6 minutes ago

He described his presentation as a follow-up to the **Google** 2009 **I/O** conference last year, in which Vic Gundotra, **Google** vice president of engineering, **...**

Two new tools for real-time web apps

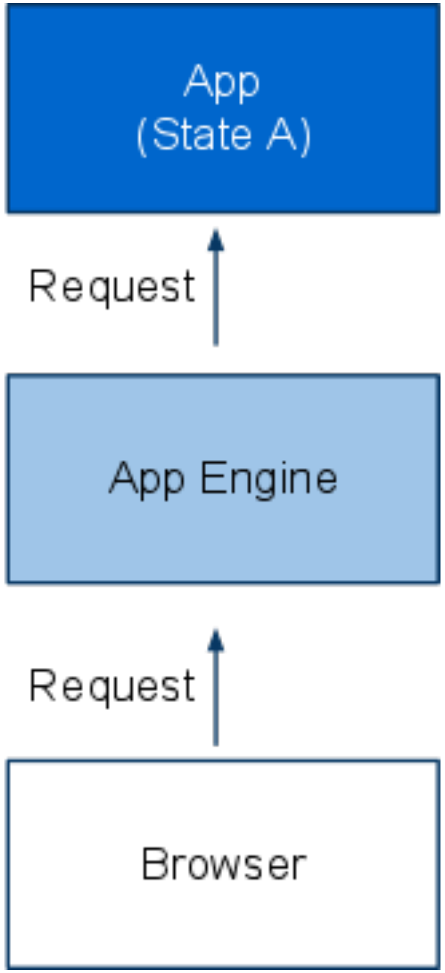
- App Engine Channel API
- Feed API v2 with Push Updates

App Engine Channel API

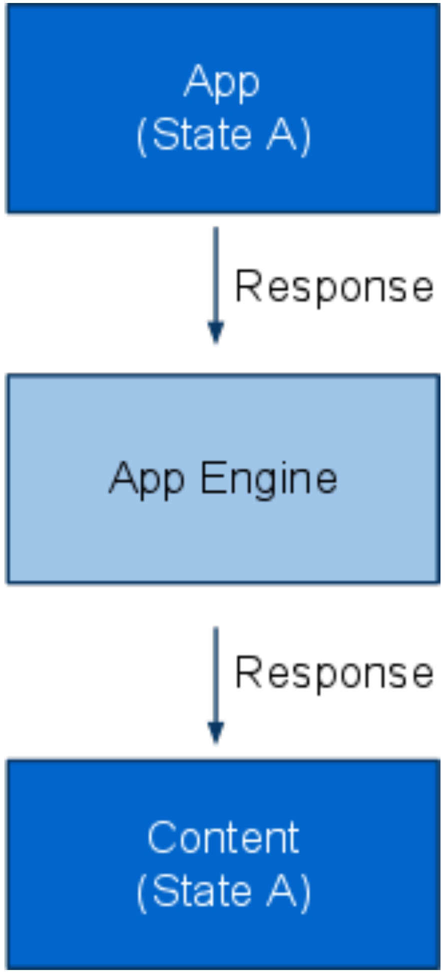
Channel API: Agenda

- The way things are now
- What the Channel API does
- How to use it
- How it works

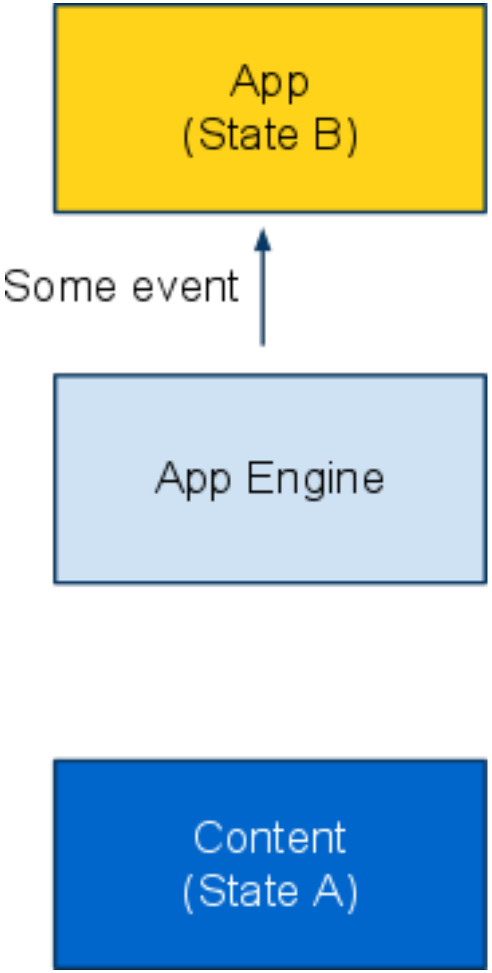
Channel API: Current model



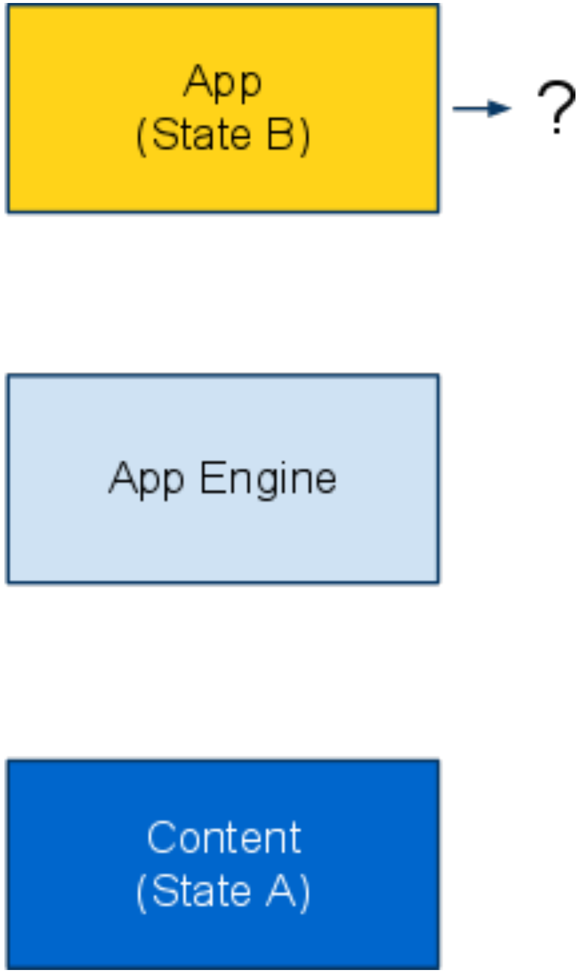
Channel API: Current model



Channel API: Current model



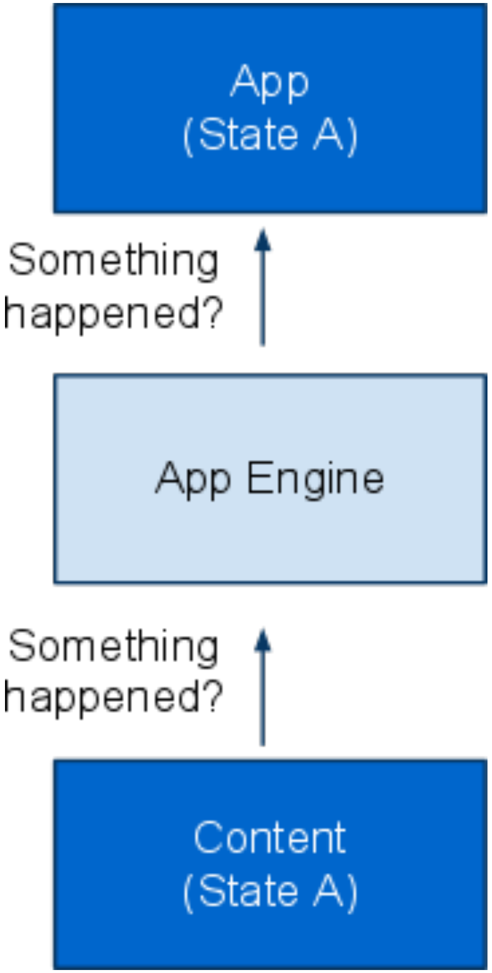
Channel API: Current model



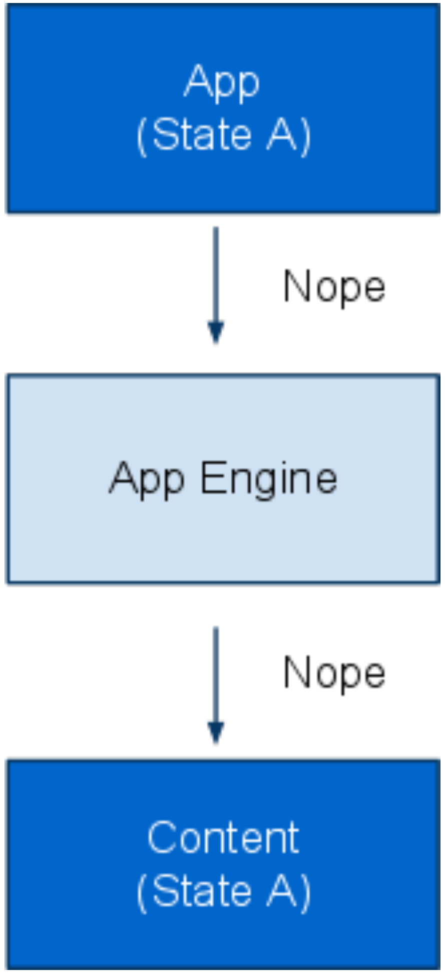
Channel API: Uh oh!

- Something changed, but I can't tell the user!
- Examples: games, chat, collaboration.
- Basically any interaction with another person or group.
- You could use polling, but...

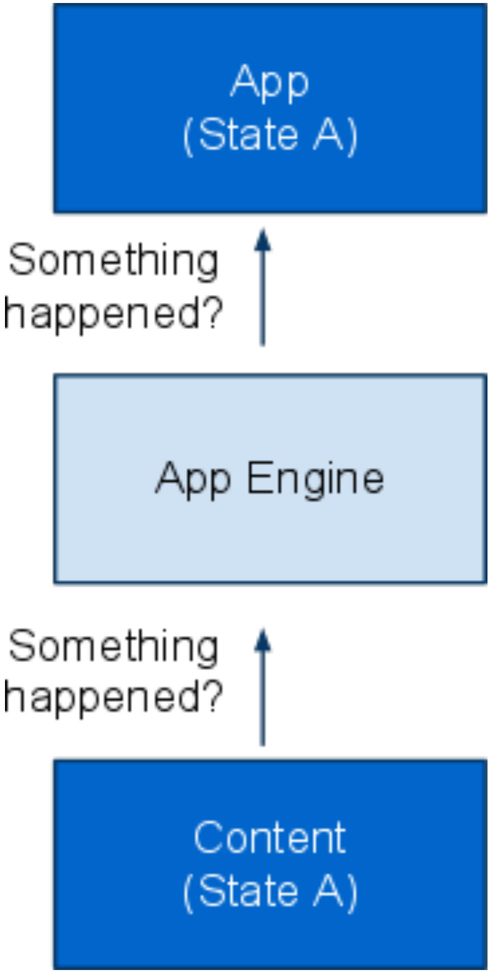
Channel API: Polling FTL



Channel API: Polling FTL



Channel API: Polling FTL



App
(State A)

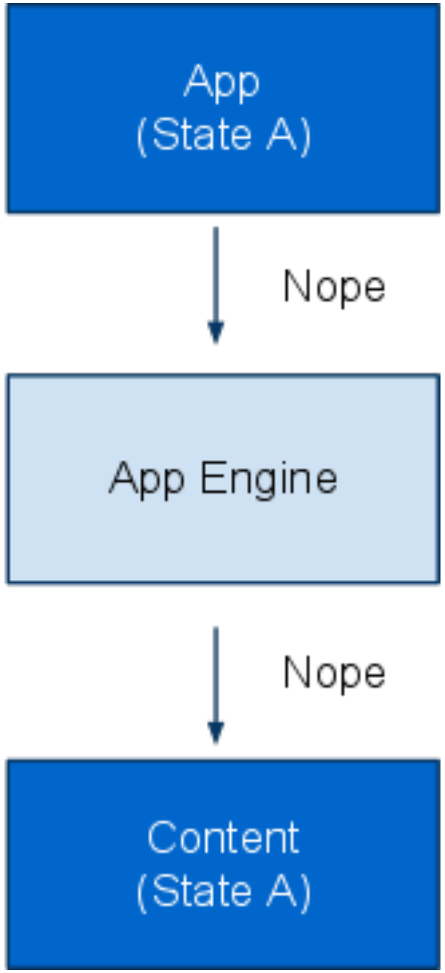
Something happened?

App Engine

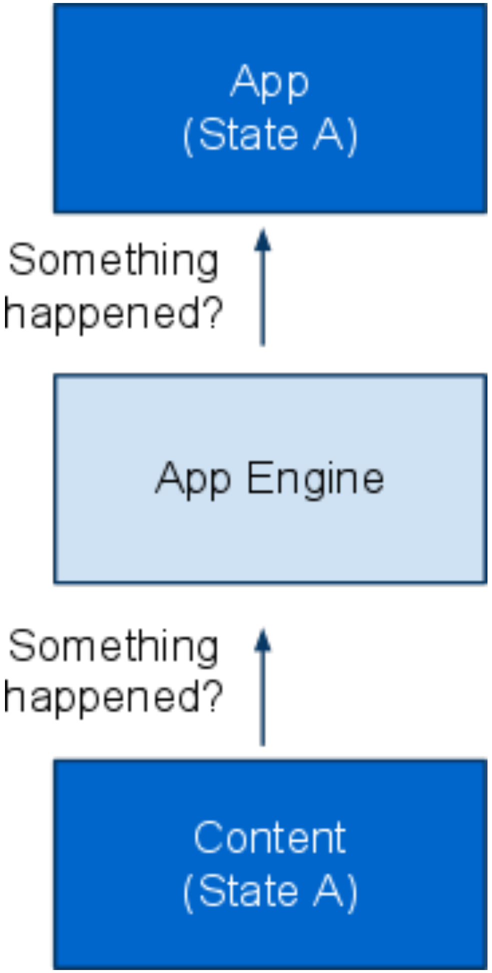
Something happened?

Content
(State A)

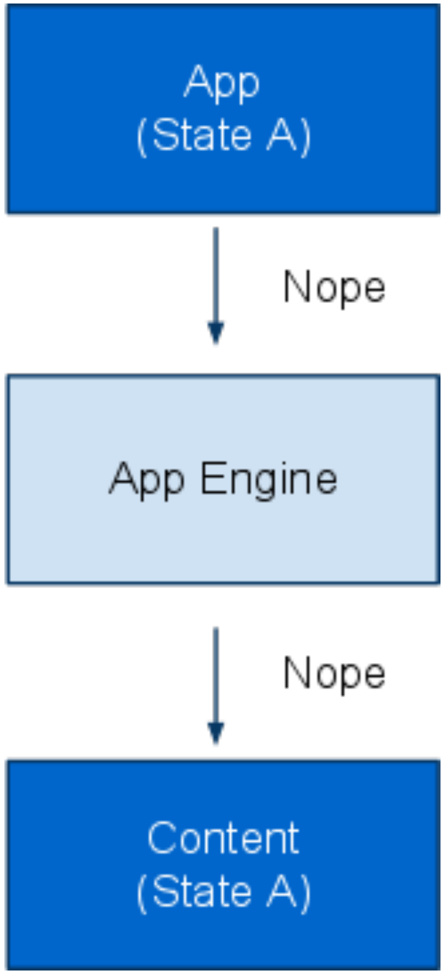
Channel API: Polling FTL



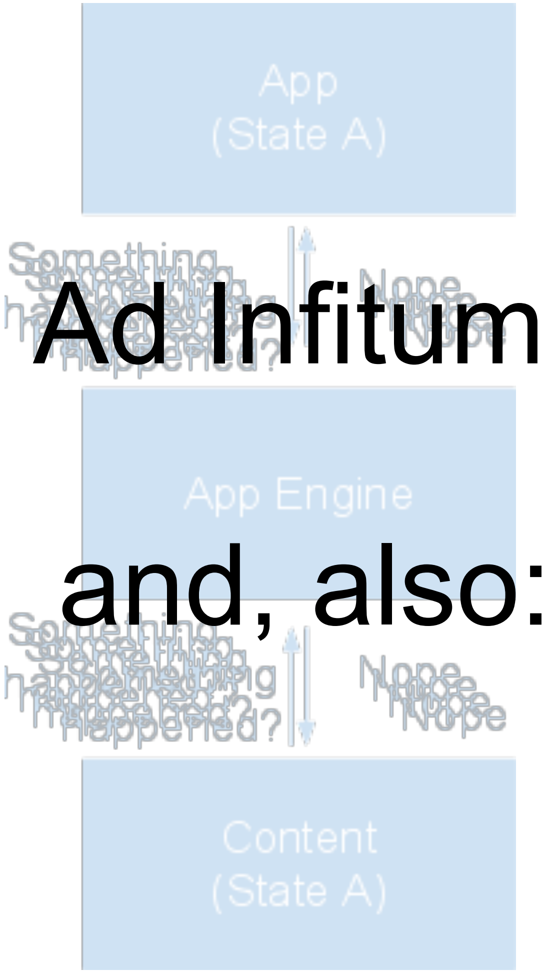
Channel API: Polling FTL



Channel API: Polling FTL



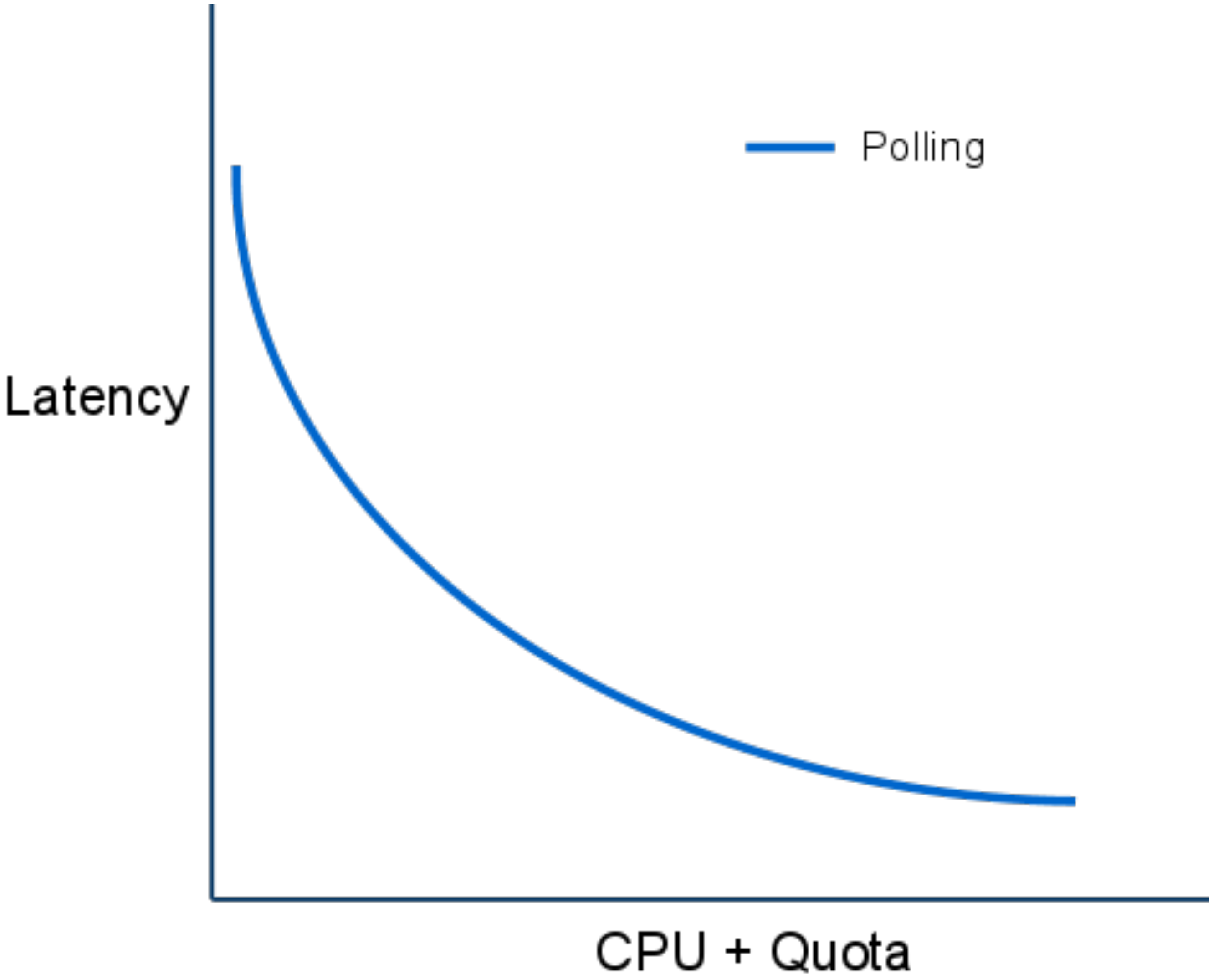
Channel API: Polling FTL



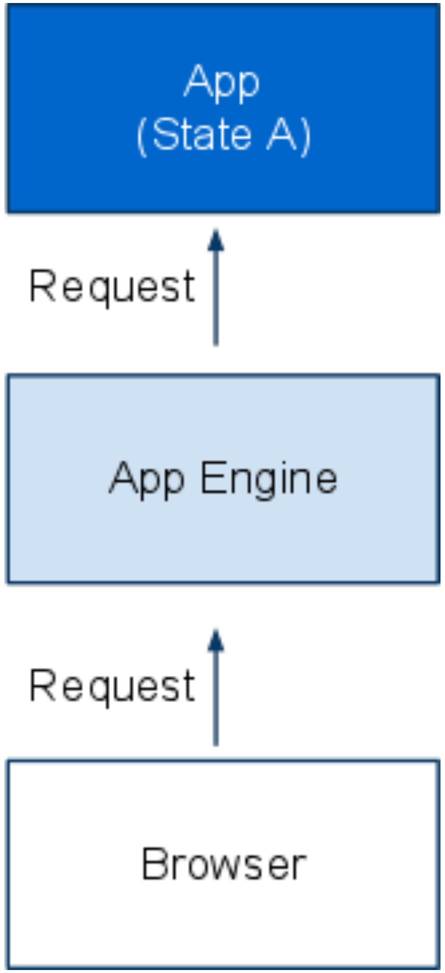
Ad Infitum!

and, also:

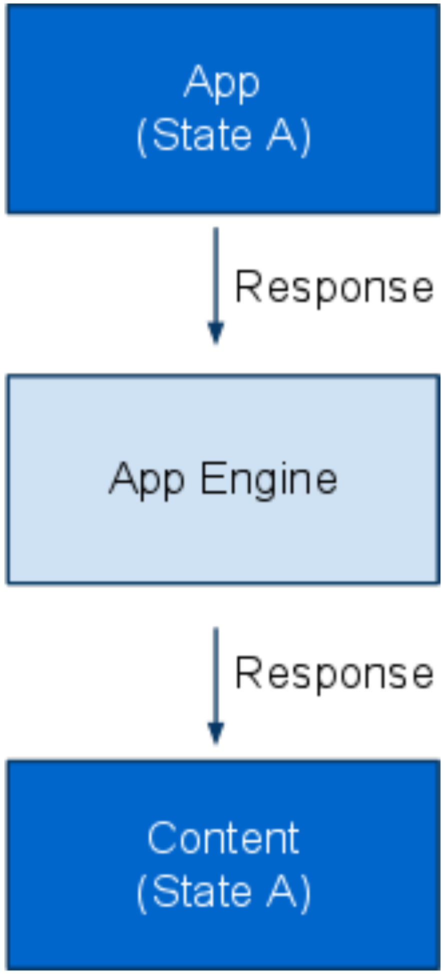
Channel API: Polling Latency vs. Cost



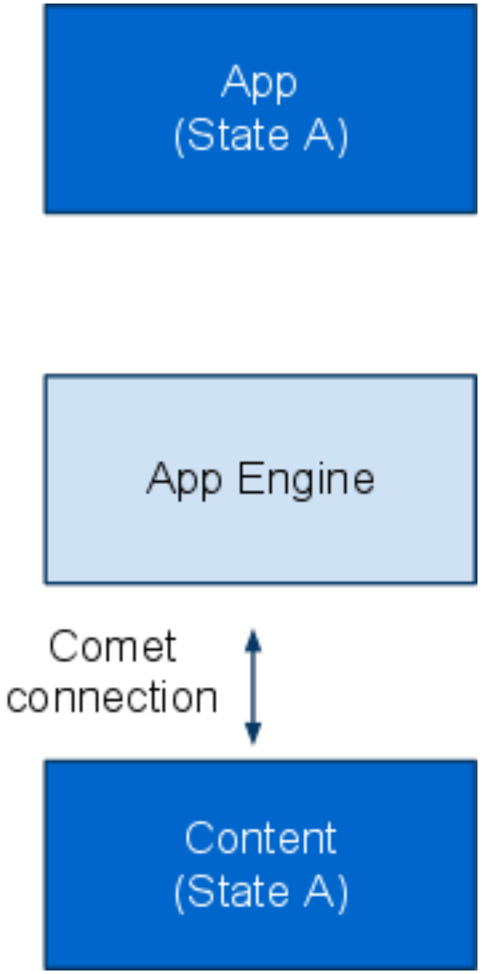
Channel API: The new way



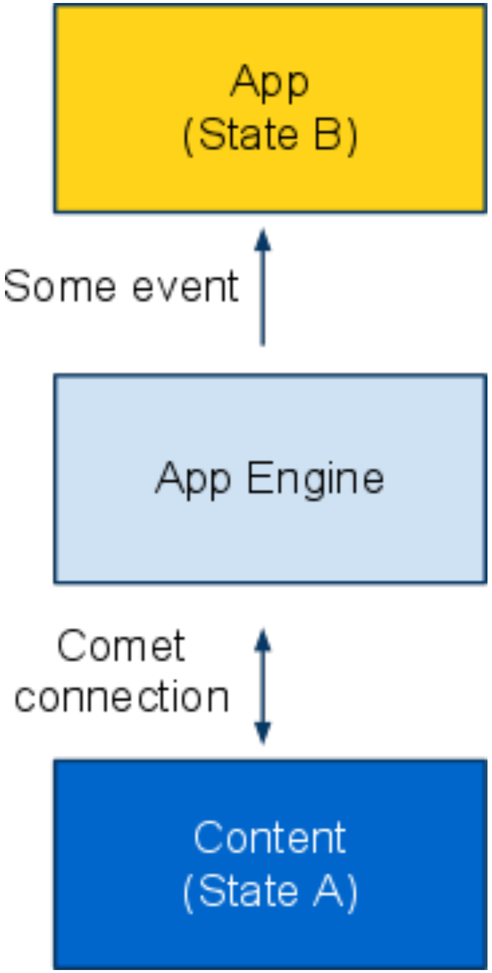
Channel API: The new way



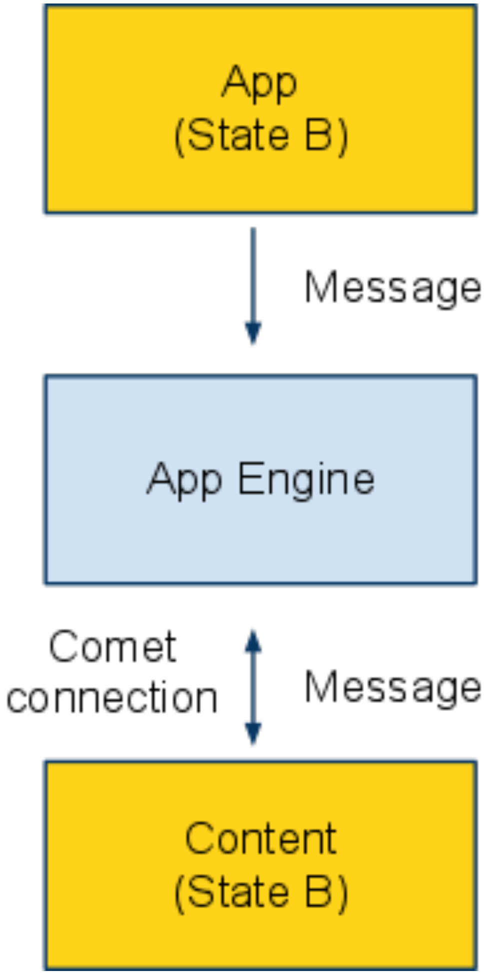
Channel API: The new way



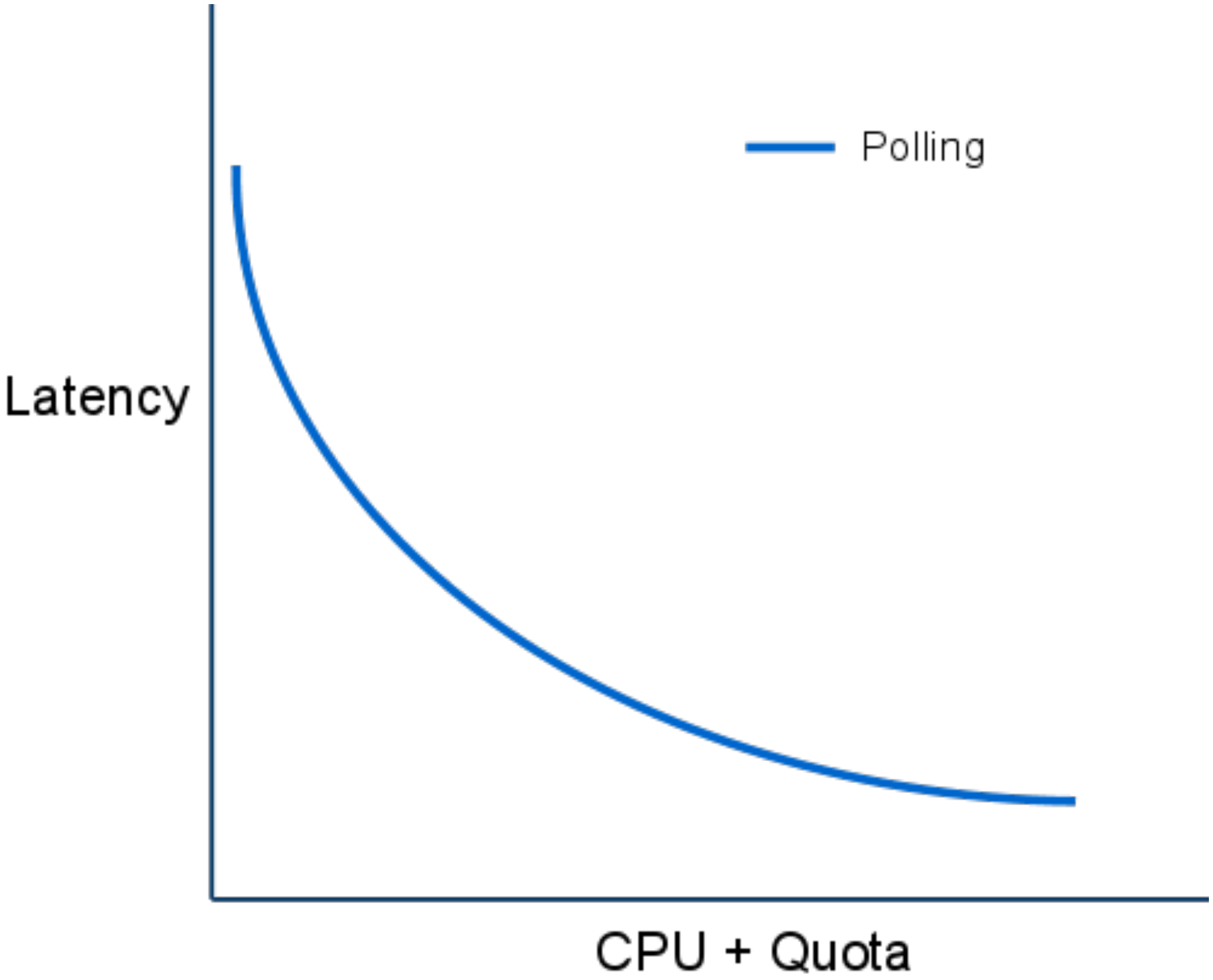
Channel API: The new way



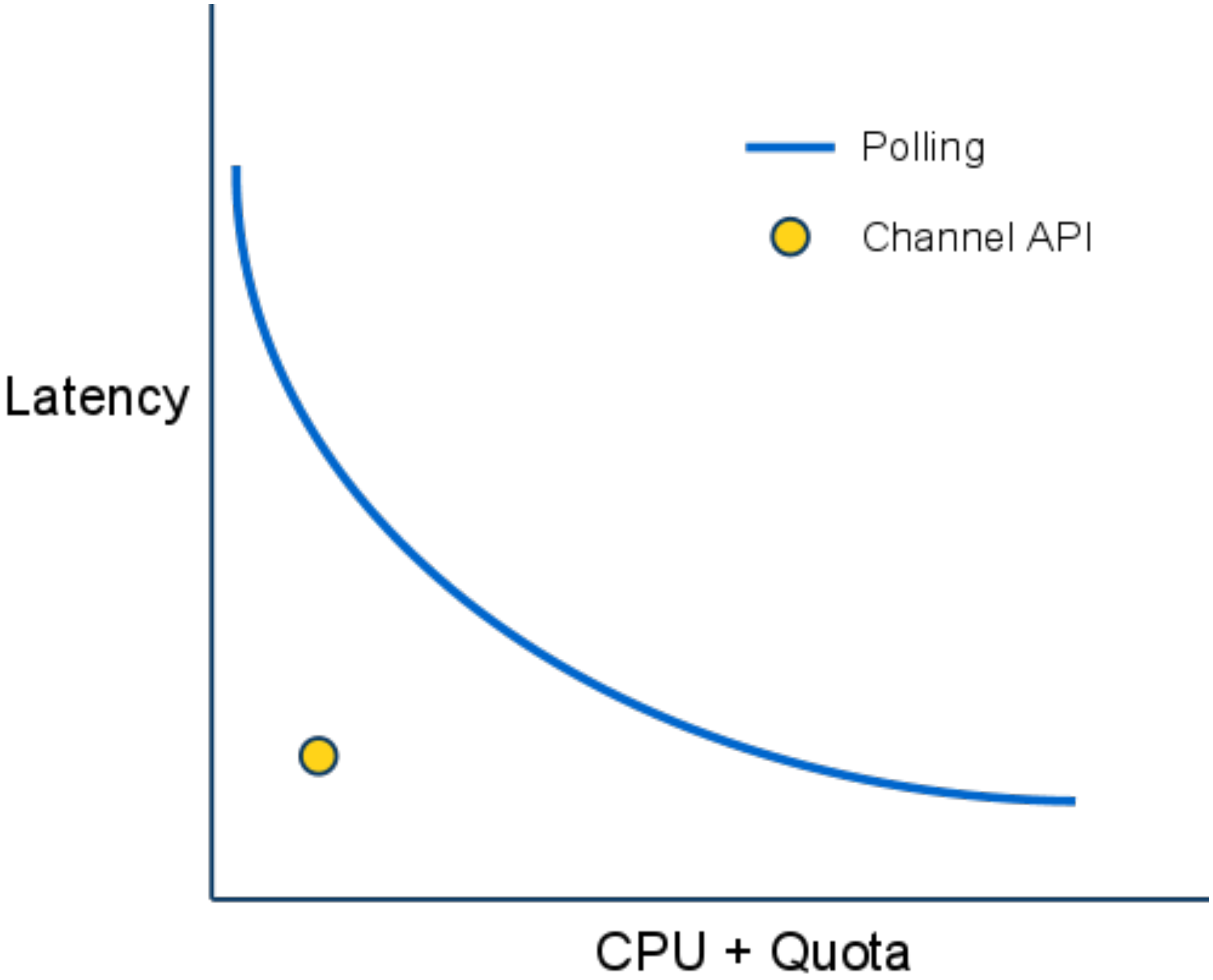
Channel API: The new way



Channel API: Easier & better than polling



Channel API: Easier & better than polling

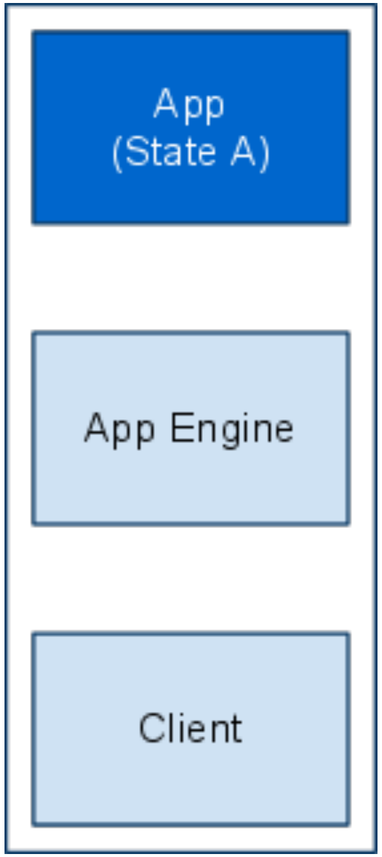


Demo time

(demo, switch browser window)

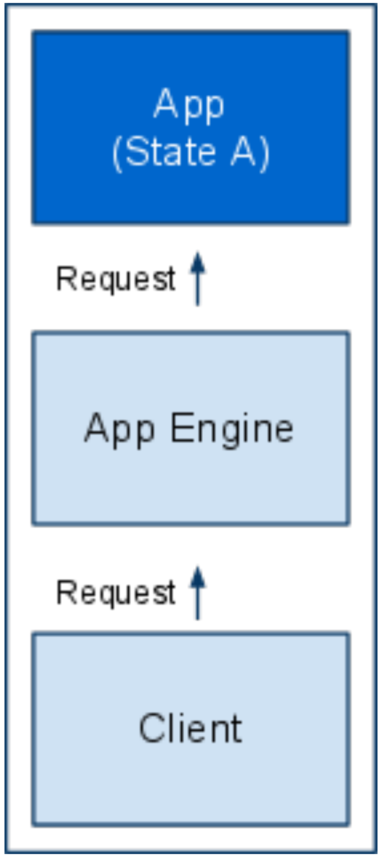
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



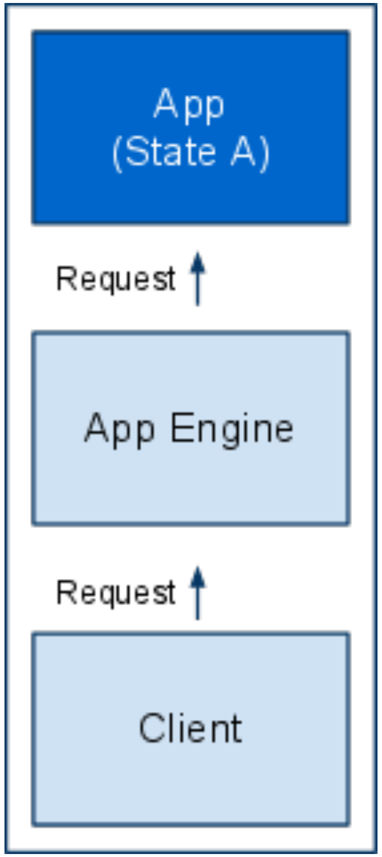
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



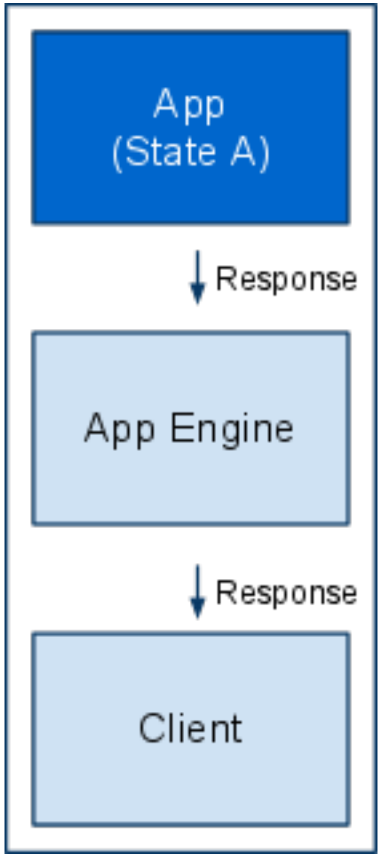
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



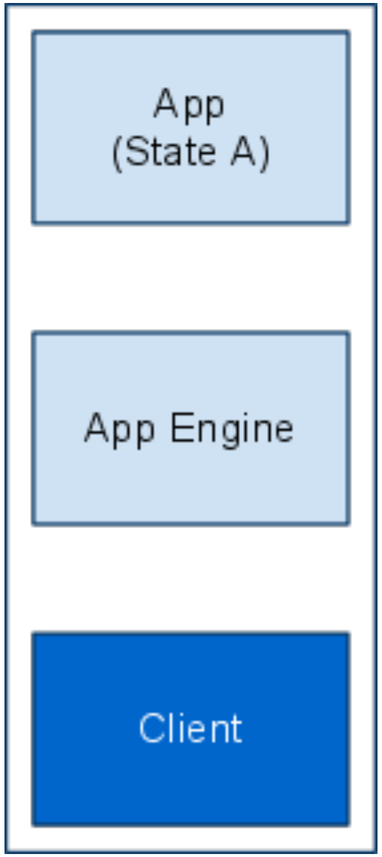
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



Channel API: How to use it

```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



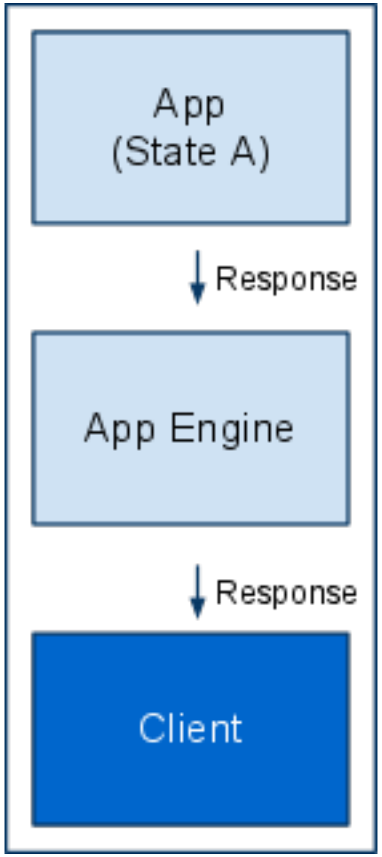
App
(State A)

App Engine

Client

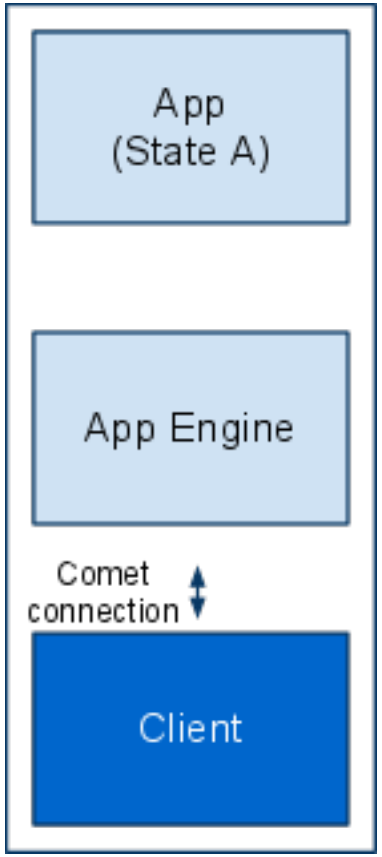
Channel API: How to use it

```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



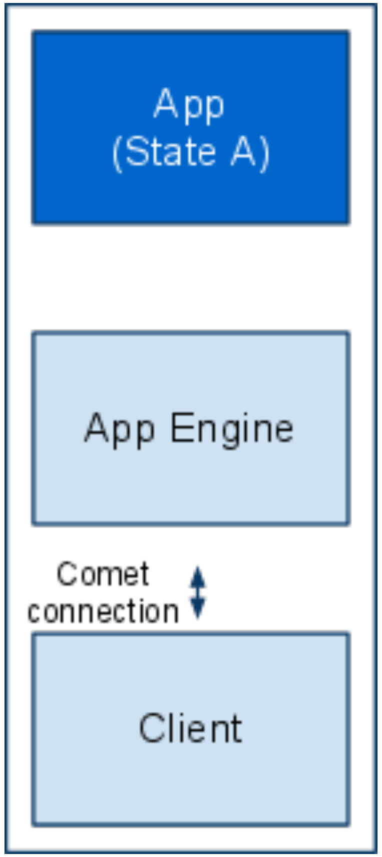
Channel API: How to use it

```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



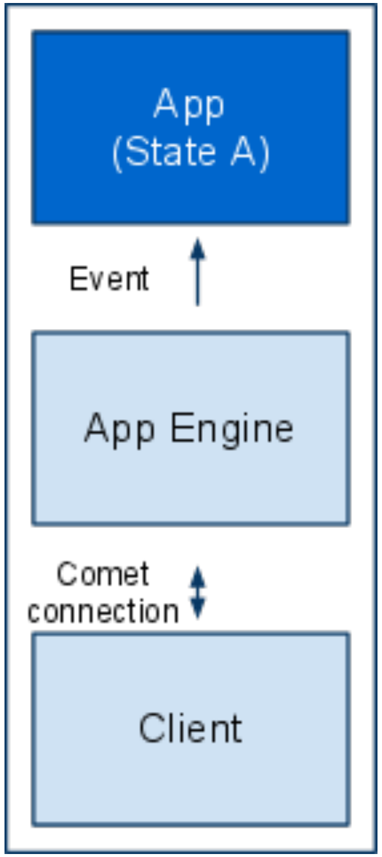
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



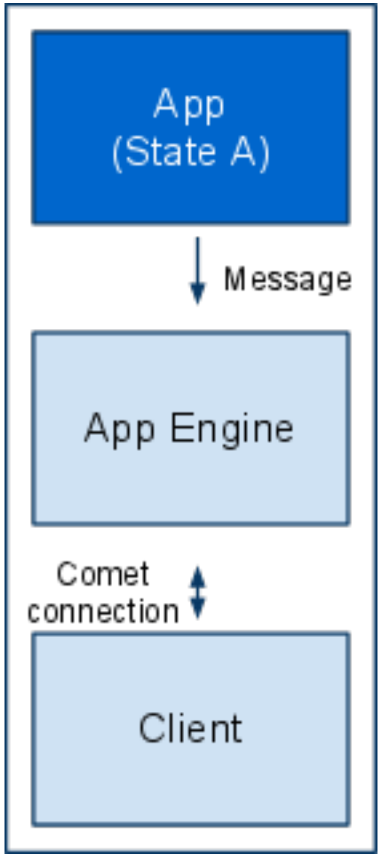
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



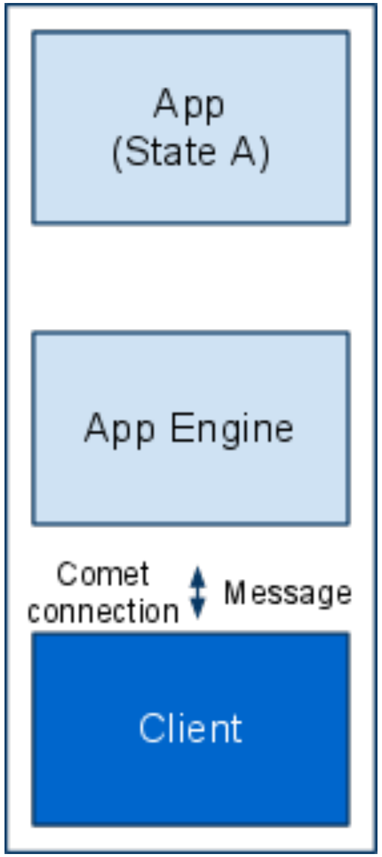
Channel API: How to use it

```
class MainPage(webapp.RequestHandler):

  def get(self):
    id = channel.create_channel(key)
    self.response.out.write(
      {'channel_id': id})


class OtherPage(webapp.RequestHandler):

  def get(self):
    # something happened!
    channel.send_message(key, 'bar')
```



Channel API: How to use it

```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



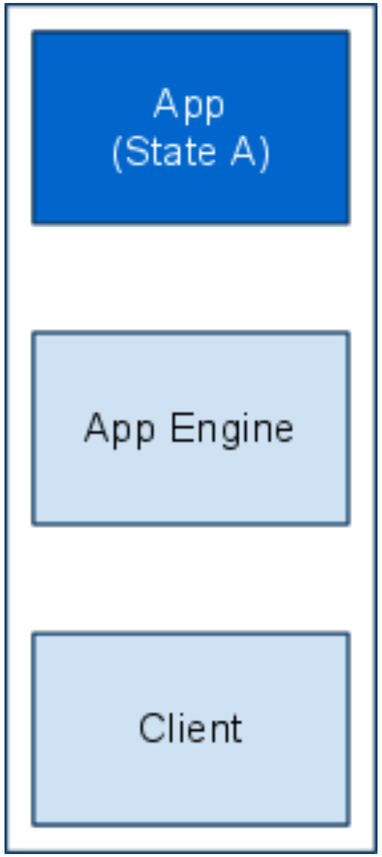
Channel API: How to use it (details)

Python API:

```
createChannel(application_key)
sendMessage(application_key, message)
```

Java API:

```
public interface ChannelService {
  String createChannel(String applicationKey);
  void sendMessage(ChannelMessage message);
}
```



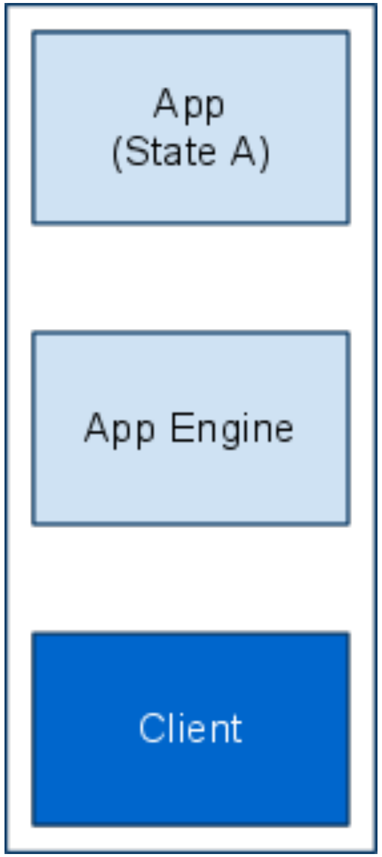
App
(State A)

App Engine

Client

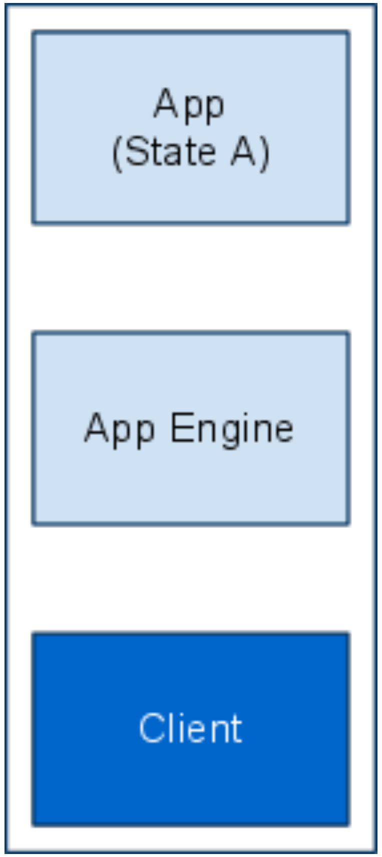
Channel API: How to use it (details)

```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



Channel API: How to use it (details)

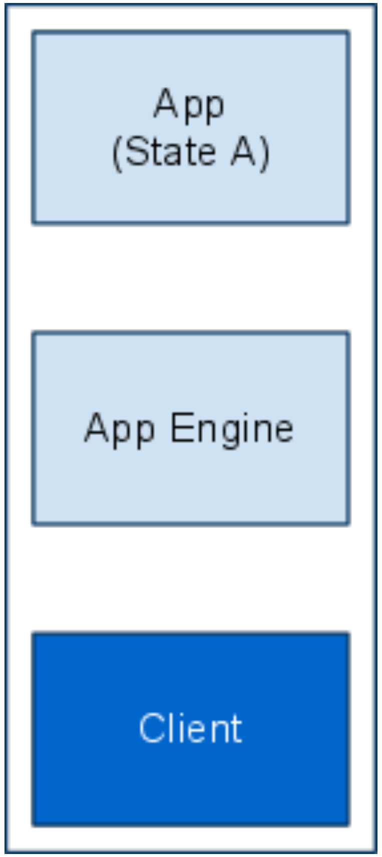
```
<script src='/_ah/channel/jsapi'></script>
<script>
  var channelId = '{{ channel_id }}';
  var channel =
    new goog.appengine.Channel(channelId);
  var socket = channel.open();
  socket.onmessage = function(evt) {
    alert(evt.data);
  };
</script>
```



Channel API: How to use it (details)

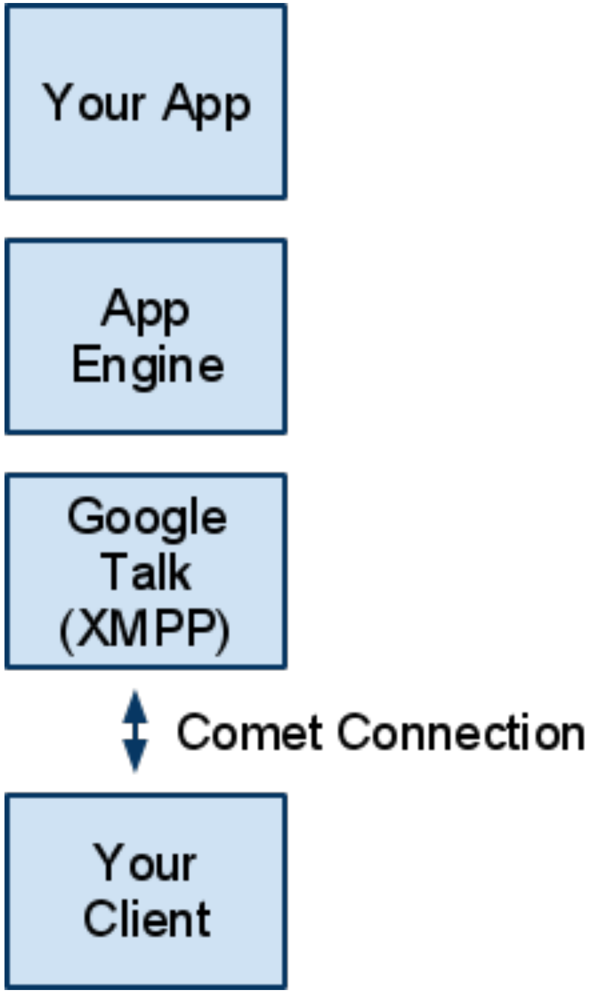
Source: http://dev.w3.org/html5/websockets/

```
interface WebSocket {
  readonly attribute DOMString url;

  // ready state
  const unsigned short CONNECTING = 0;
  const unsigned short OPEN = 1;
  const unsigned short CLOSING = 2;
  const unsigned short CLOSED = 3;
  readonly attribute unsigned short readyState;
  readonly attribute unsigned long bufferedAmount;

  // networking
           attribute Function onopen;
           attribute Function onmessage;
           attribute Function onerror;
           attribute Function onclose;
  boolean send(in DOMString data);
  void close();
};
```



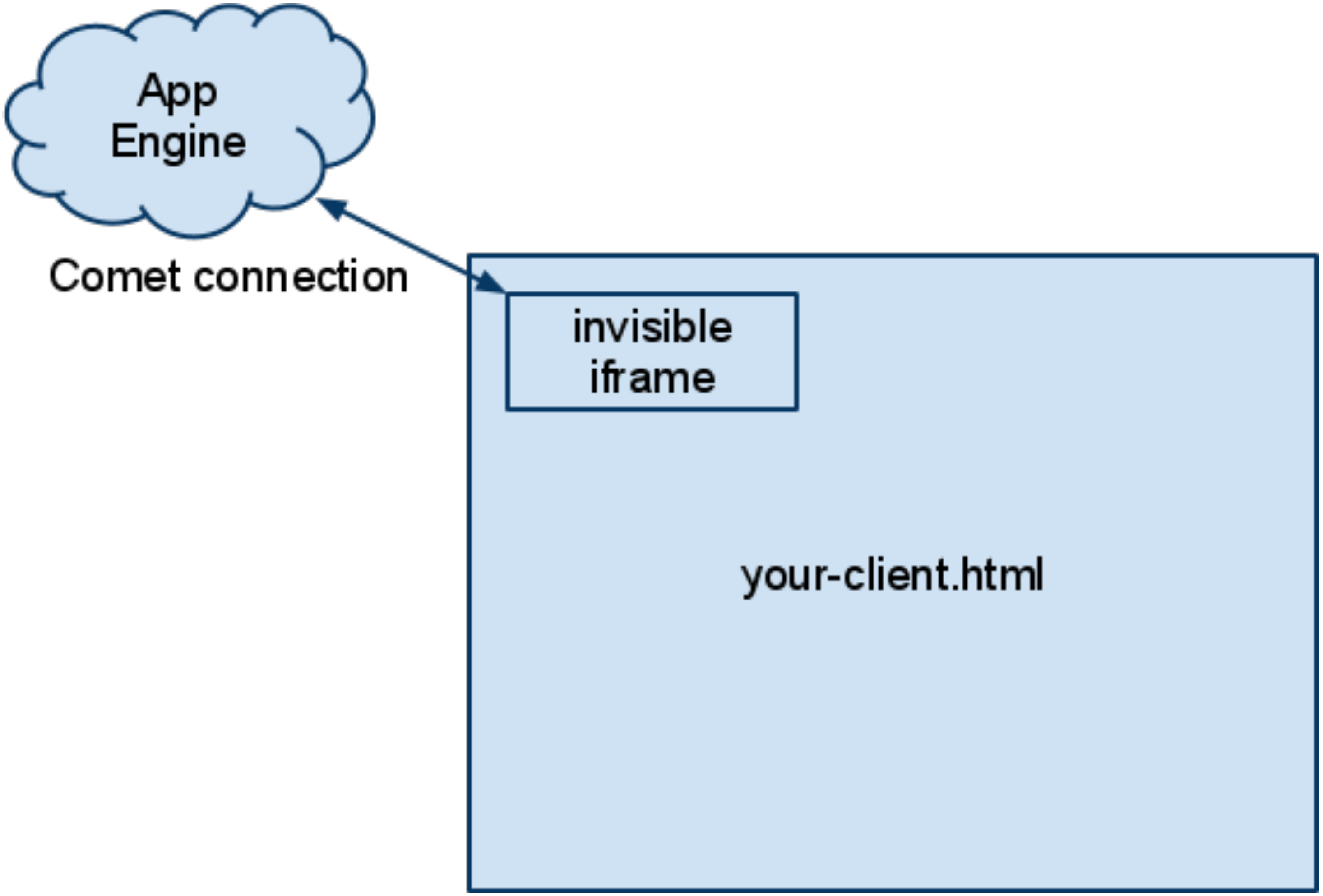
Implementation Details

- Using XMPP "under the covers" for message delivery



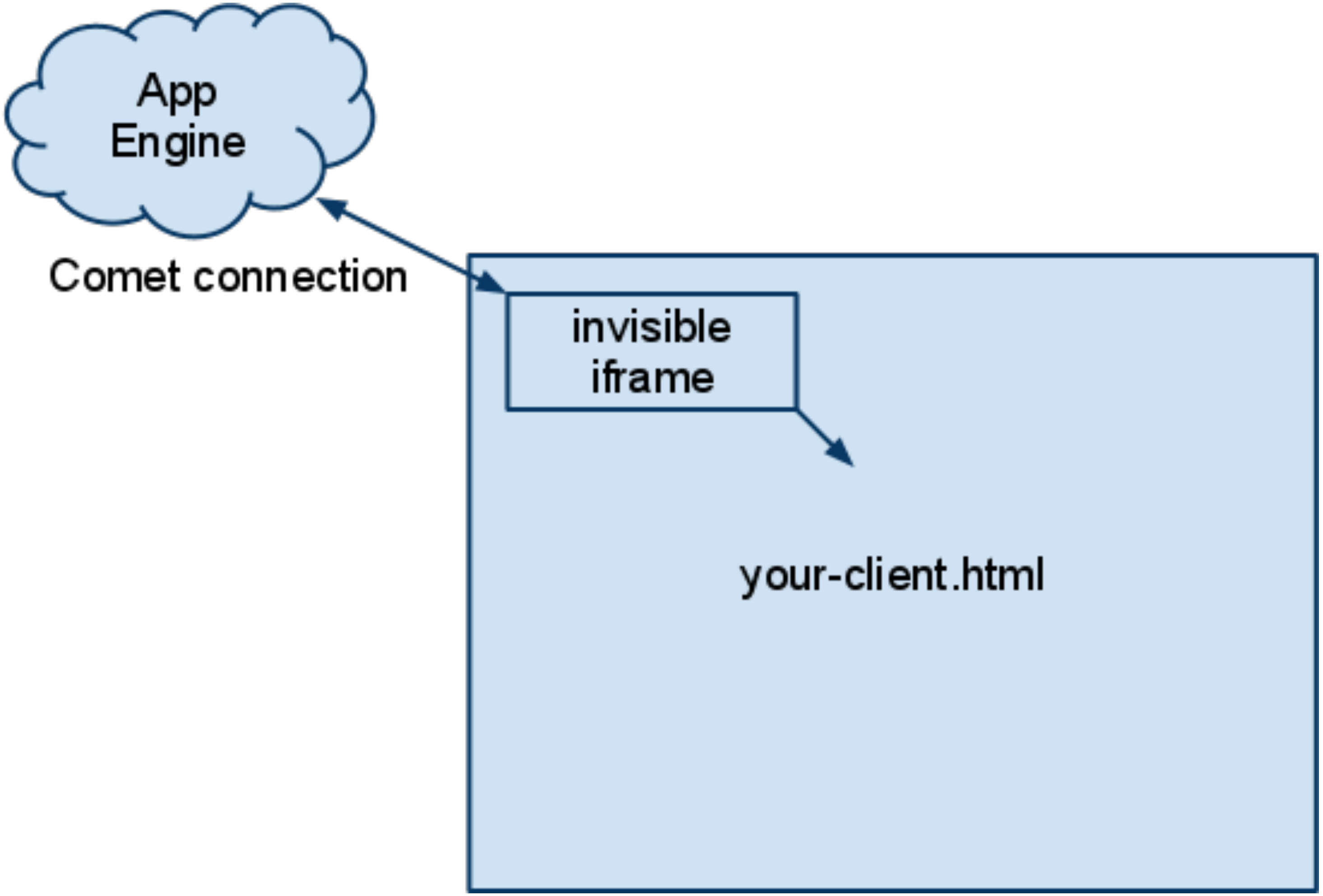
Implementation Details

- Client uses embedded iframe with connection to App Engine



Implementation Details

- Messages are relayed to client via XPC



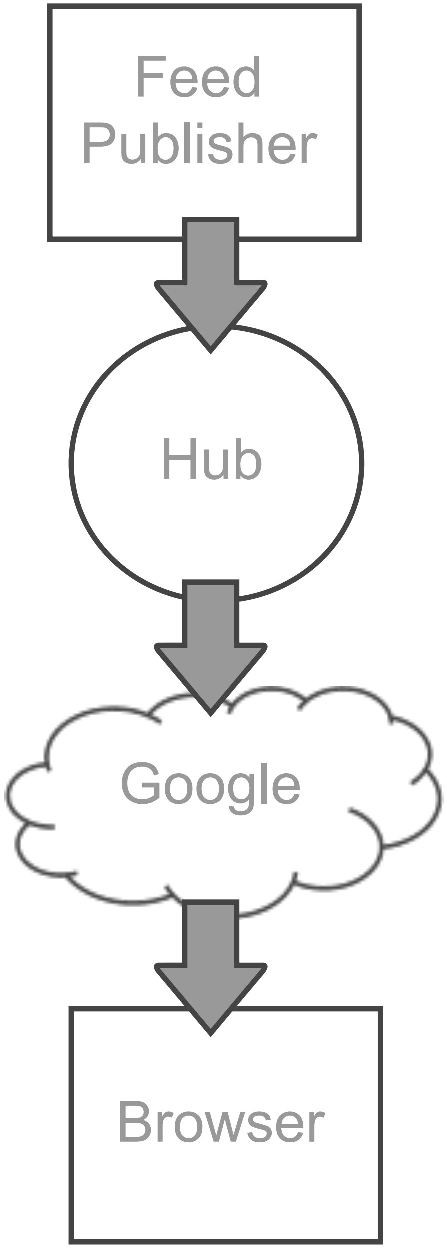
Conclusion

- Currently available to trusted testers, broadly available soon.

Feed API v2 with Push Updates

Feed API v2: Agenda

- Feed API v1 versus v2
- Loading in Feed API v1
- Subscribing in Feed API v2
- How updates are pushed
- Advanced uses of Feed API v2



Feed Publisher

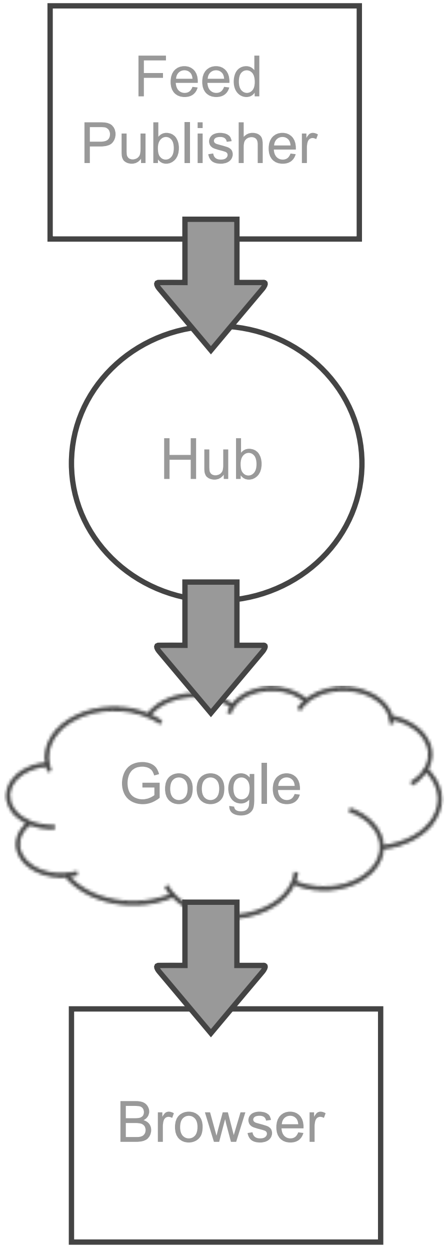
Hub

Google

Browser

Feed API v1 versus v2

- Feed API v1 **loads** current feed contents
- Feed API v2 **subscribes** to feed updates*
    - Supports any PubSubHubbub feed



* Feed API v2 can load feeds too, but that's nothing new

Loading in Feed API v1

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  // TODO: Do something with the result
}
```

Loading in Feed API v1

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  // TODO: Do something with the result
}
```

Loading in Feed API v1

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  // TODO: Do something with the result
}
```

Loading in Feed API v1

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  // TODO: Do something with the result
}
```

An example of feed XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
  <channel>
    <title>Google I/O</title>
    <link>http://www.google.com</link>
    <description>Awesome!</description>
    <item>
      <title>This session</title>
      <link>http://ajax.googleapis.com</link>
      <description>Amazing!</description>
    </item>
    <item>
      <title>Your next session</title>
      <link>http://code.google.com</link>
      <description>Awe-inspiring!</description>
    </item>
  </channel>
</rss>
```

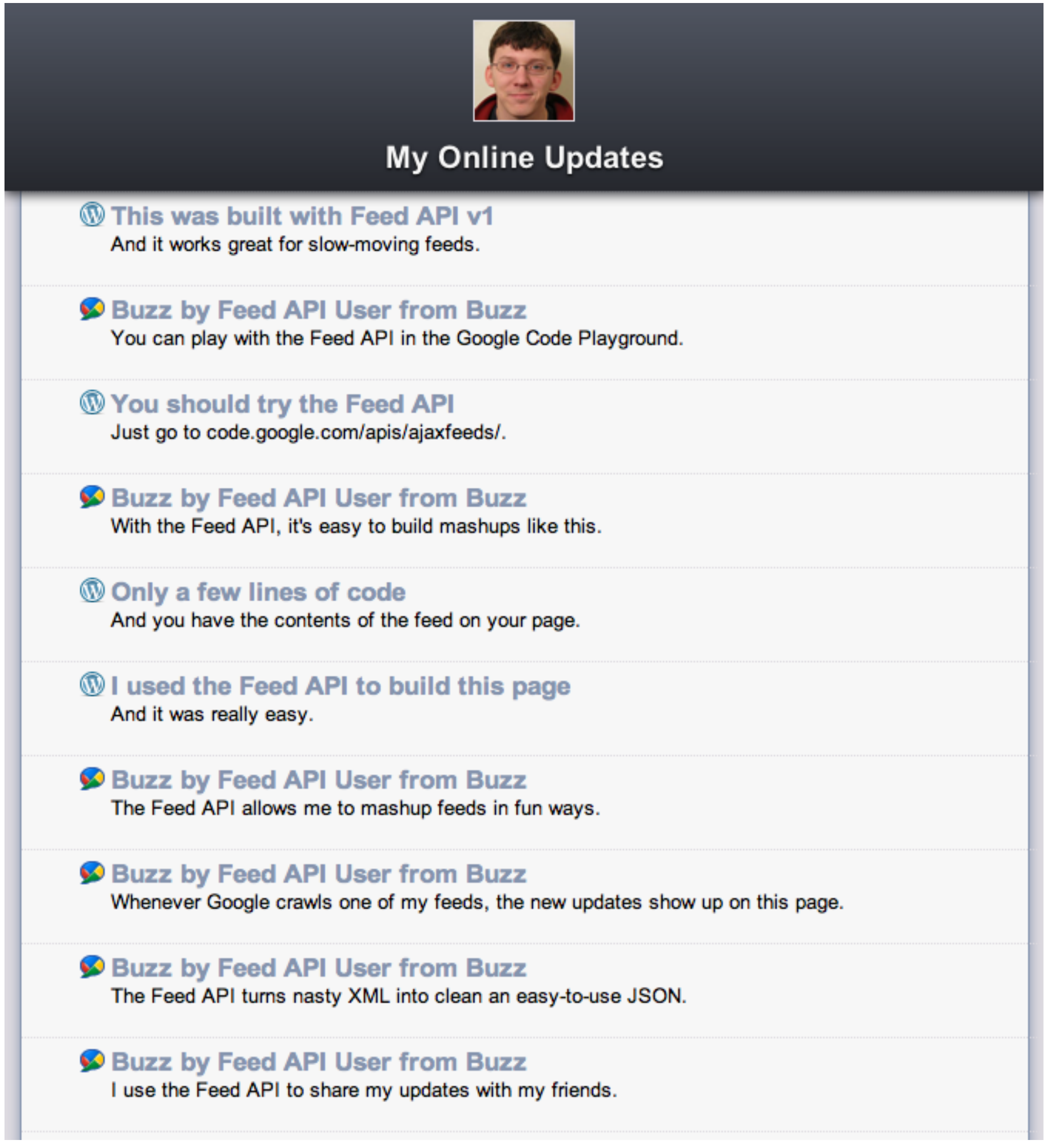
The JSON result format

```
{
  feed: {
    title: "Google I/O",
    link: "http://www.google.com",
    description: "Awesome!",
    entries: [{
      title: "This session",
      link: "http://ajax.googleapis.com",
      content: "Amazing!"
    },{
      title: "Your next session",
      link: "http://code.google.com",
      content: "Awe-inspiring!"
    }]
  }
}
```

Using the JSON result

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

A demo built with Feed API v1



My Online Updates

This was built with Feed API v1
And it works great for slow-moving feeds.

Buzz by Feed API User from Buzz
You can play with the Feed API in the Google Code Playground.

You should try the Feed API
Just go to code.google.com/apis/ajaxfeeds/.

Buzz by Feed API User from Buzz
With the Feed API, it's easy to build mashups like this.

Only a few lines of code
And you have the contents of the feed on your page.

I used the Feed API to build this page
And it was really easy.

Buzz by Feed API User from Buzz
The Feed API allows me to mashup feeds in fun ways.

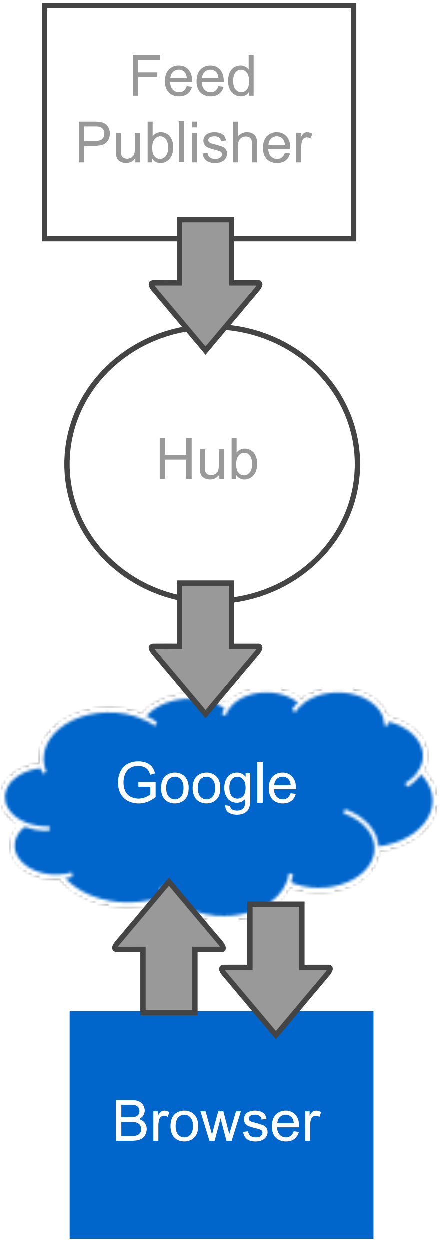
Buzz by Feed API User from Buzz
Whenever Google crawls one of my feeds, the new updates show up on this page.

Buzz by Feed API User from Buzz
The Feed API turns nasty XML into clean an easy-to-use JSON.

Buzz by Feed API User from Buzz
I use the Feed API to share my updates with my friends.

How about Feed API v1 with polling?

- Multiple levels of cache prevent updates from showing up very quickly
- Poll too much and your traffic looks abusive, may be blocked
- Poll too little and you might as well not



Loading in Feed API v1

```
google.load("feeds", "1");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.load(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

Subscribing in Feed API v2

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

Subscribing in Feed API v2

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

Subscribing in Feed API v2

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

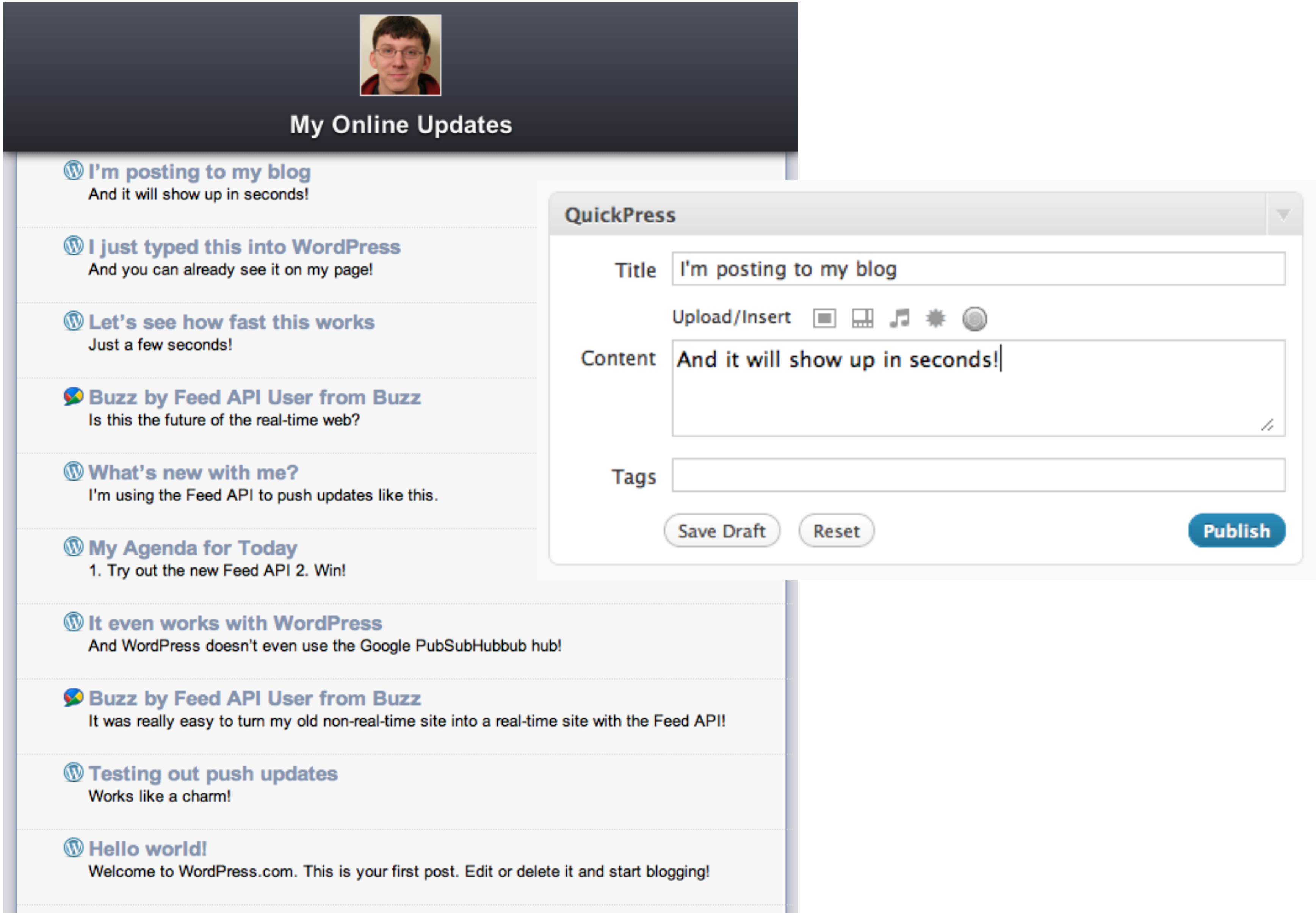
Subscribing in Feed API v2

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

Subscribing in Feed API v2

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```

A demo built with Feed API v2



My Online Updates

I'm posting to my blog
And it will show up in seconds!

I just typed this into WordPress
And you can already see it on my page!

Let's see how fast this works
Just a few seconds!

Buzz by Feed API User from Buzz
Is this the future of the real-time web?

What's new with me?
I'm using the Feed API to push updates like this.

My Agenda for Today
1. Try out the new Feed API 2. Win!

It even works with WordPress
And WordPress doesn't even use the Google PubSubHubbub hub!

Buzz by Feed API User from Buzz
It was really easy to turn my old non-real-time site into a real-time site with the Feed API!

Testing out push updates
Works like a charm!

Hello world!
Welcome to WordPress.com. This is your first post. Edit or delete it and start blogging!

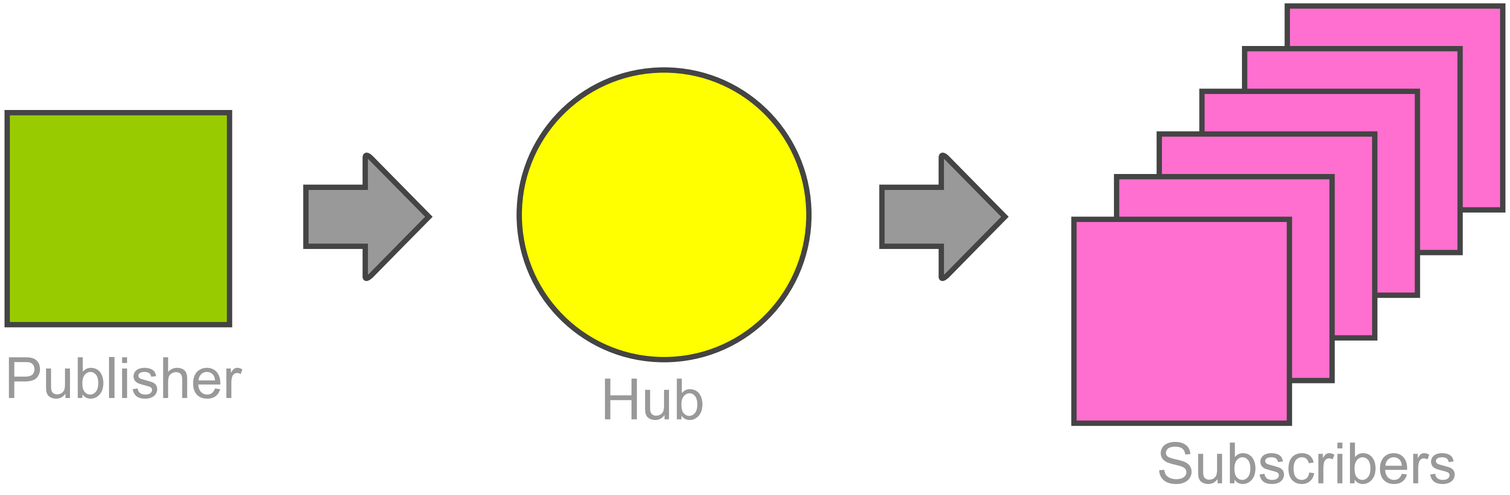
QuickPress

Title: I'm posting to my blog

Content: And it will show up in seconds!

Tags:

PubSubHubbub



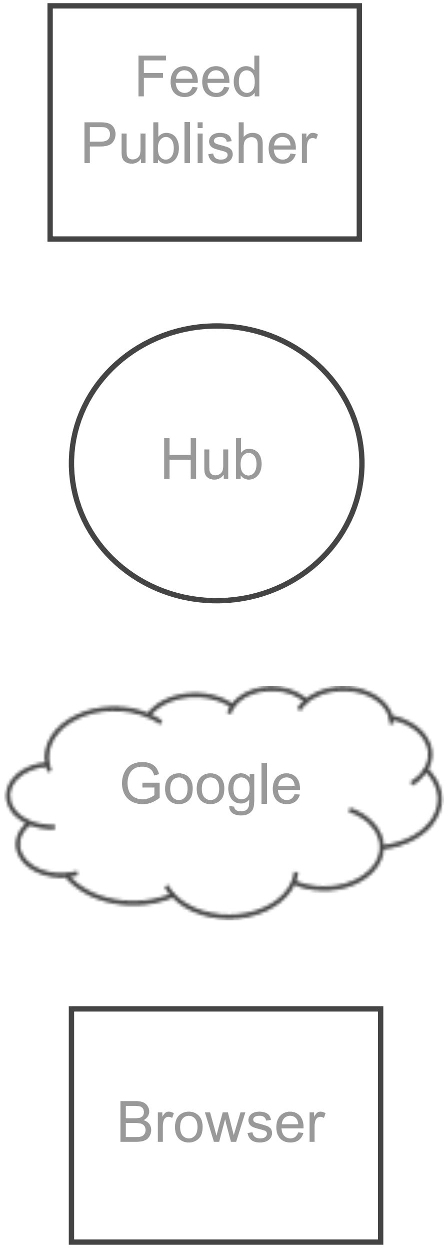
code.google.com/p/pubsubhubbub

A PubSubHubbub enabled feed

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <atom:link rel="hub" href="http://examplehub.com"/>
    <title>Google I/O</title>
    <link>http://www.google.com</link>
    <description>Awesome!</description>
    <item>
      <title>This session</title>
      <link>http://ajax.googleapis.com</link>
      <description>Amazing!</description>
    </item>
    <item>
      <title>Your next session</title>
      <link>http://code.google.com</link>
      <description>Awe-inspiring!</description>
    </item>
  </channel>
</rss>
```

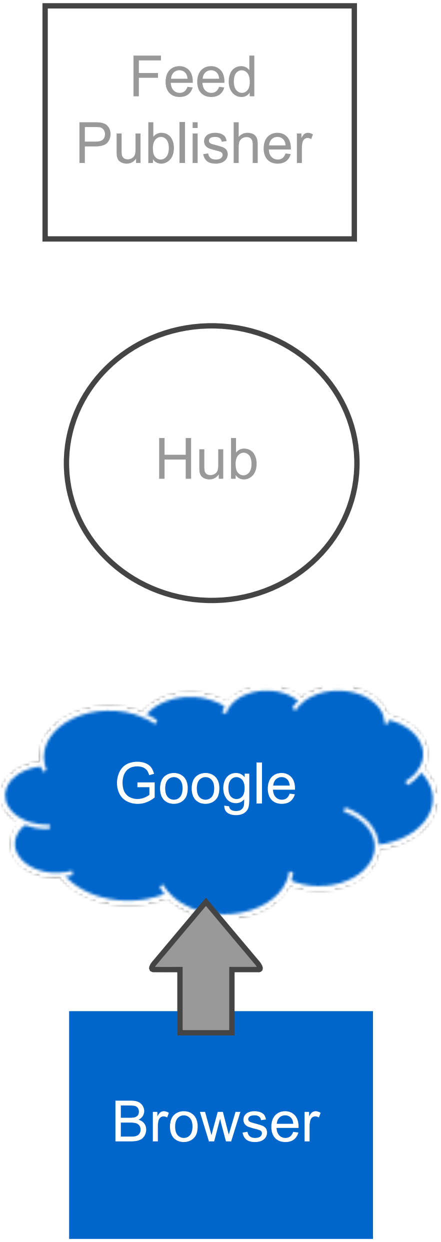
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



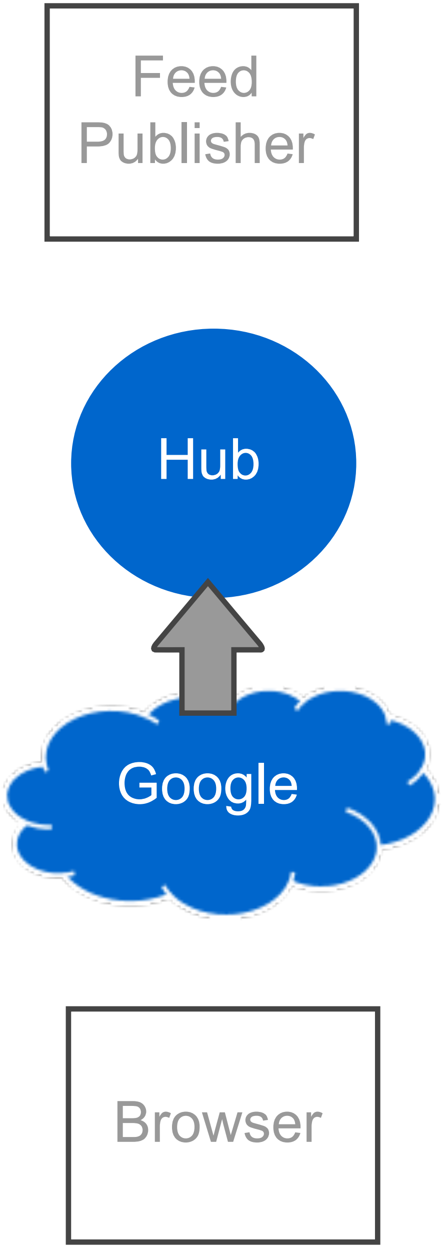
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



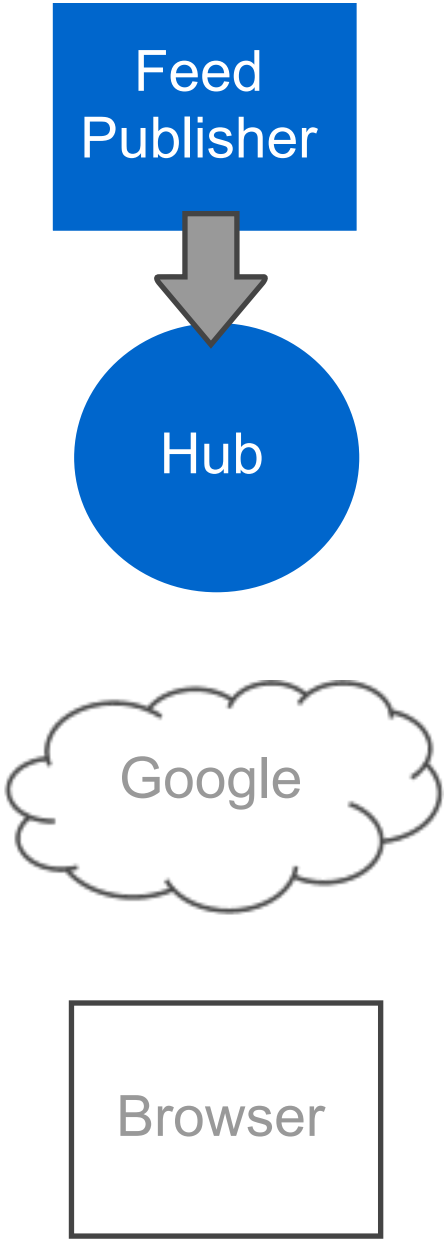
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



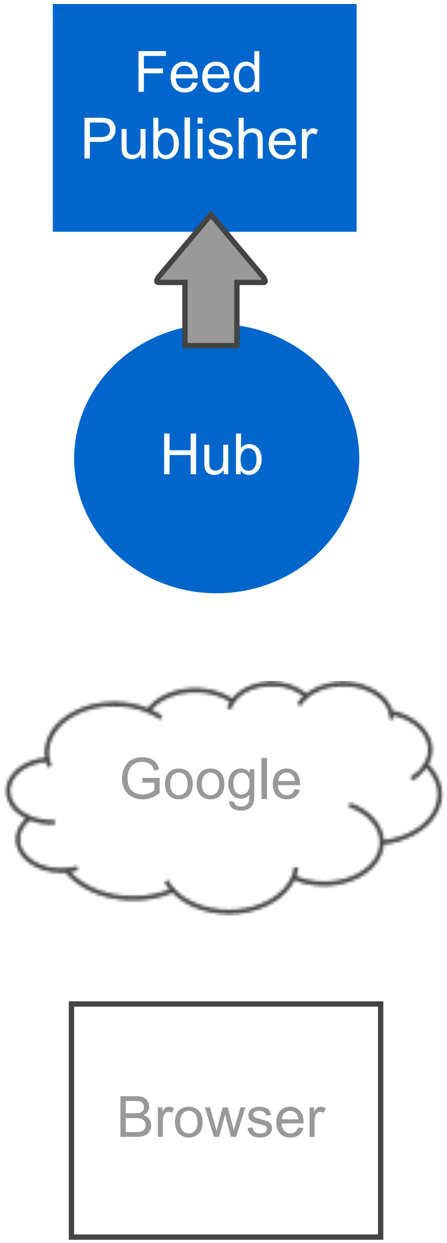
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



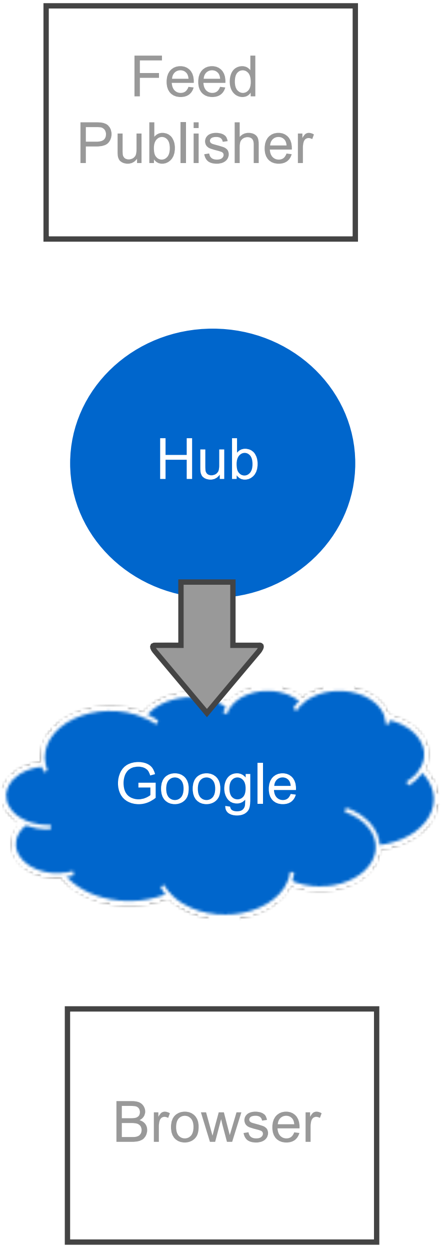
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



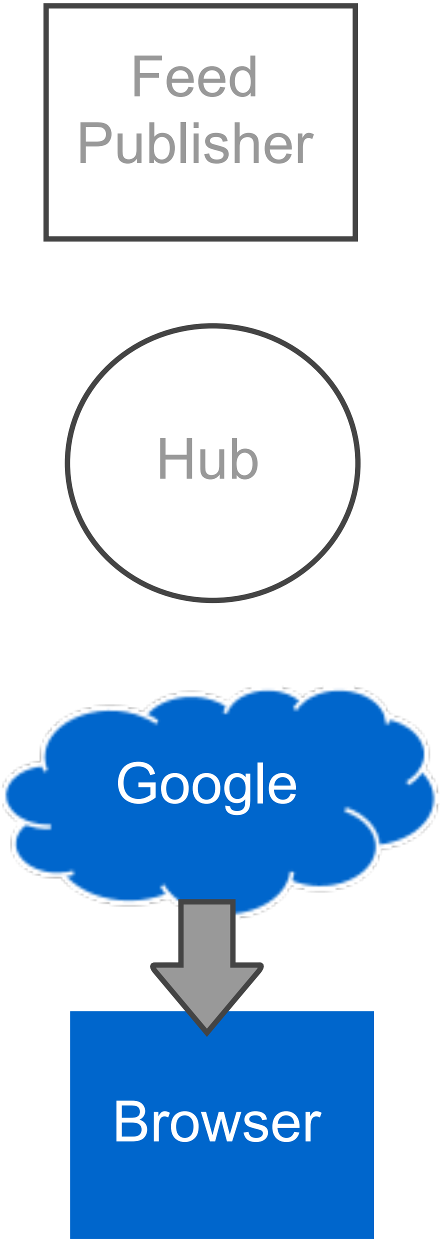
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



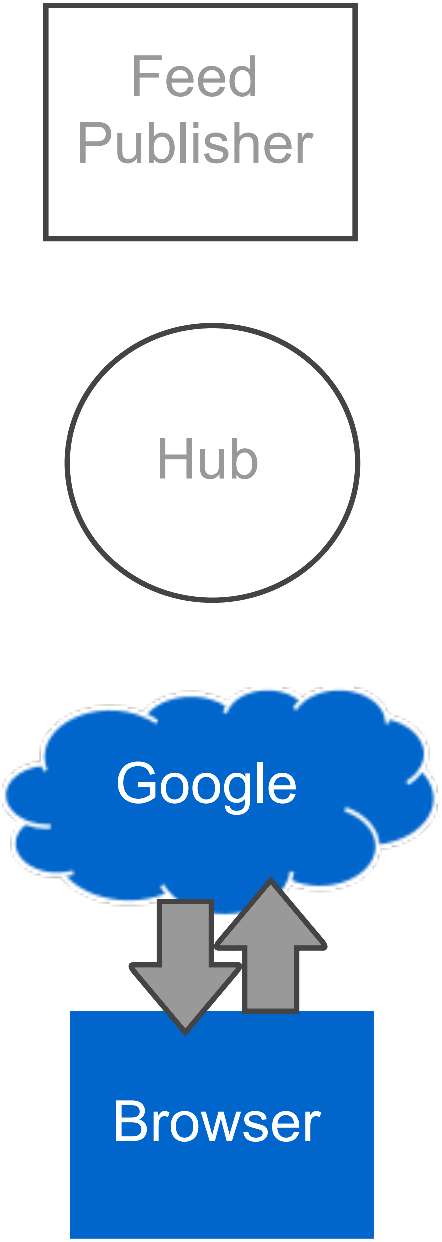
How updates are pushed

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



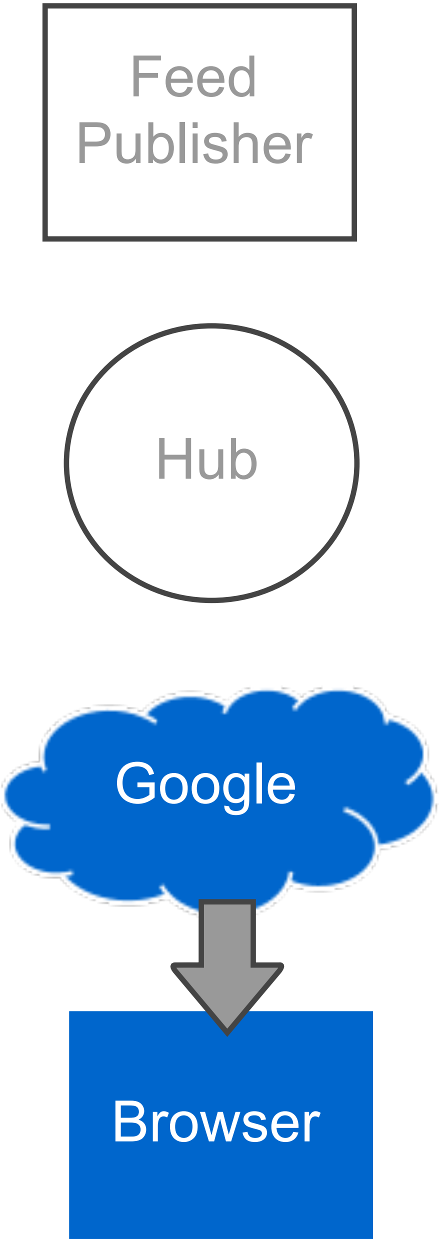
Advanced: Historical entries

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed.includeHistoricalEntries(20);
  feed.subscribe(doSomethingWithFeedContents);
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



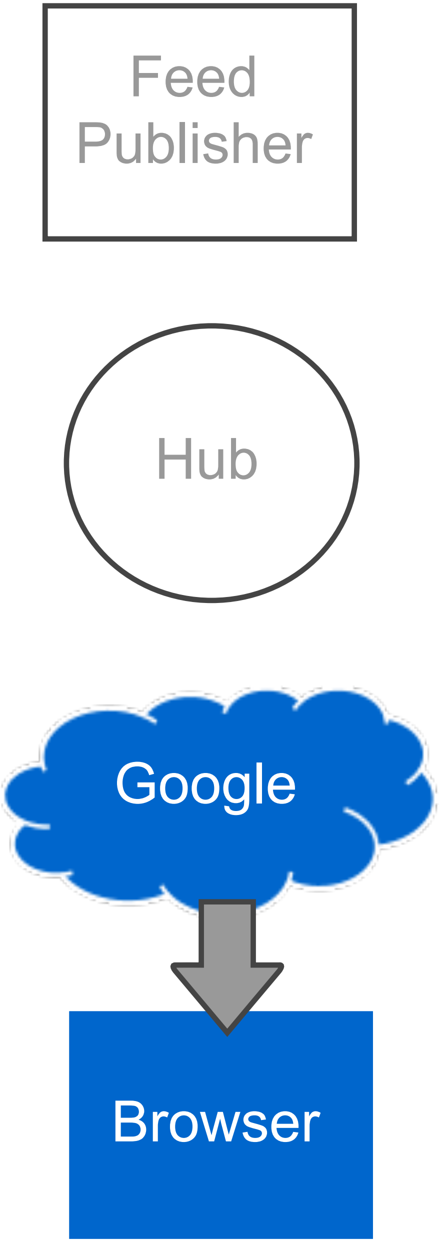
Advanced: Multiple feeds

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed1 = new google.feeds .push.Feed(
     "http://googleblog.blogspot.com/atom.
xml");
  feed1.subscribe(updateOnFeed1);
  var feed2 = new google.feeds.push.Feed(
     "http://googlecode.blogspot.com/atom.
xml");
  feed2.subscribe(updateOnFeed2);
}

function updateOnFeed1(result) {
  // TODO: Do something with the result
}

function updateOnFeed2(result) {
  // TODO: Do something with the result
}
```



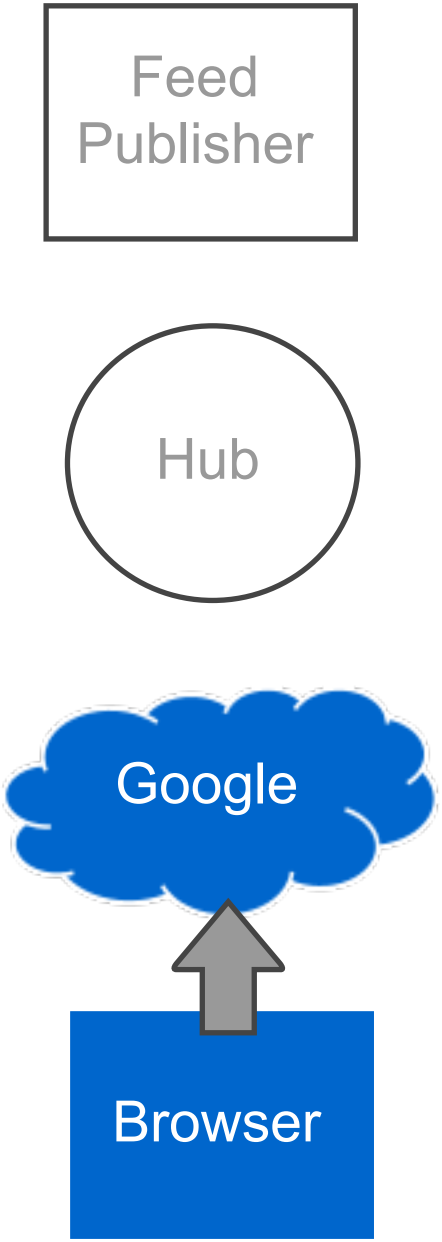
Advanced: Channel events

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  google.feeds.push.setOnBindCallback(bound);
  google.feeds.push.setOnUnbindCallback
(unbound);
  var feed = new google.feeds .push.Feed(
      "http://googleblog.blogspot.com/atom.xml");
  feed.subscribe(doSomethingWithFeedContents);
}

function bound() {
  alert("Bound channel to Google!");
}

function unbound() {
  alert("Unbound channel to Google!");
}
```



Advanced: Unsubscribing

```
google.load("feeds", "2");
google.setOnLoadCallback(onLoadOfFeedApi);

function onLoadOfFeedApi() {
  var feed = new google.feeds.push.Feed(
      "http://googleblog.blogspot.com/atom.
xml");
  feed.subscribe(doSomethingWithFeedContents);
  feed.unsubscribe();
}

function doSomethingWithFeedContents(result) {
  var entries = result.feed.entries;
  for (var i = 0; i < entries.length; i++) {
    alert(entries[i].title);
  }
}
```



Getting started

code.google.com/apis/feed/push

Getting started

code.google.com/playground

Conclusion

- App Engine Channel API
  - A channel to push from App Engine to the browser
- Feed API v2 with Push Updates
  - Any PubSubHubbub feed pushed to the browser

More info...

- App Engine Channel API
  - Docs coming soon
- Feed API v2 with Push Updates
  - code.google.com/apis/feed/push
  - code.google.com/apis/ajax/playground/?exp=feeds

View live notes and ask questions about this session on Google Wave

http://bit.ly/dc6QxD