
Google™
11
I/O


Google™
11

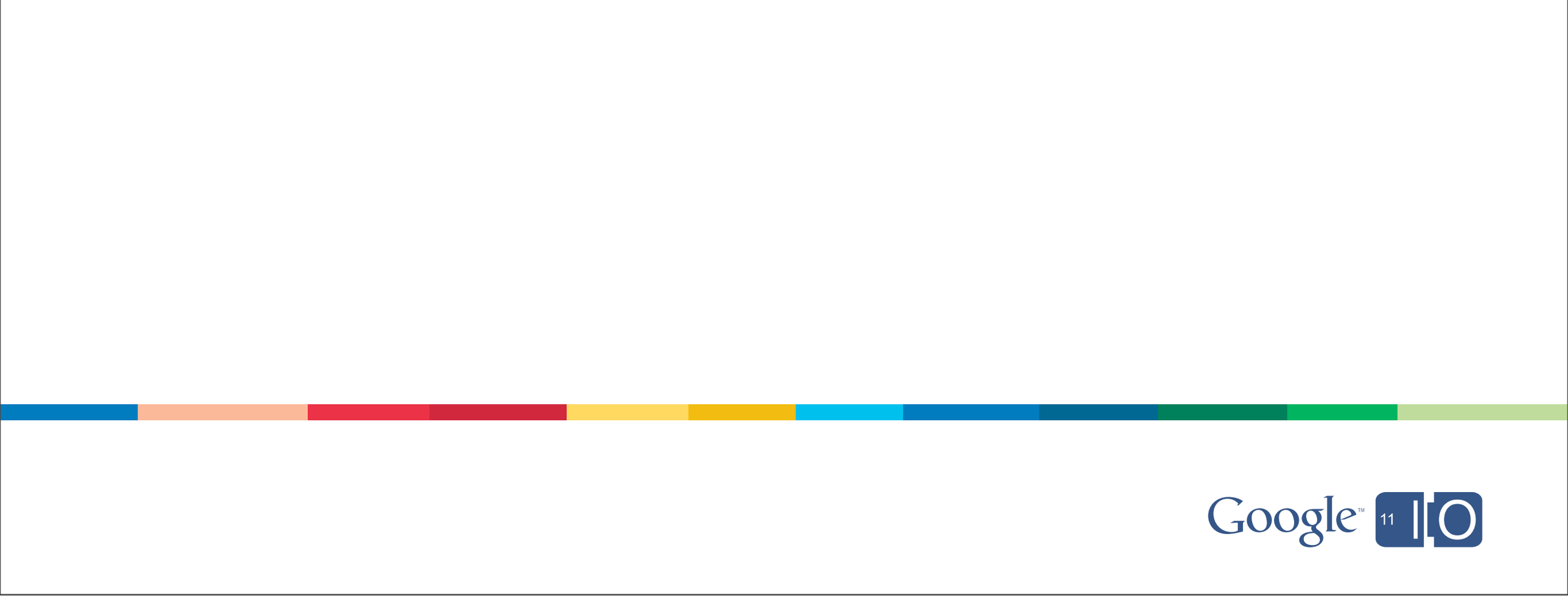
Bring the Cloud to Your IDE with GPE

Rajeev Dayal (rdayal@google.com)
May 10, 2011

Session Feedback - http://goo.gl/l7Z2y
Hashtags - #io2011 #DevTools

Agenda

- Introduction
- Getting your code from Google Project Hosting
- Using Google APIs in your project
- Deploying your project to App Engine
- Quick Bits: Other Cool Features
- Wrap-up


Google™ 11

Introduction

What is the Google Plugin for Eclipse?

- A collection of Eclipse plugins
  - works with Eclipse 3.4 - Eclipse 3.6 (3.7 imminent!)

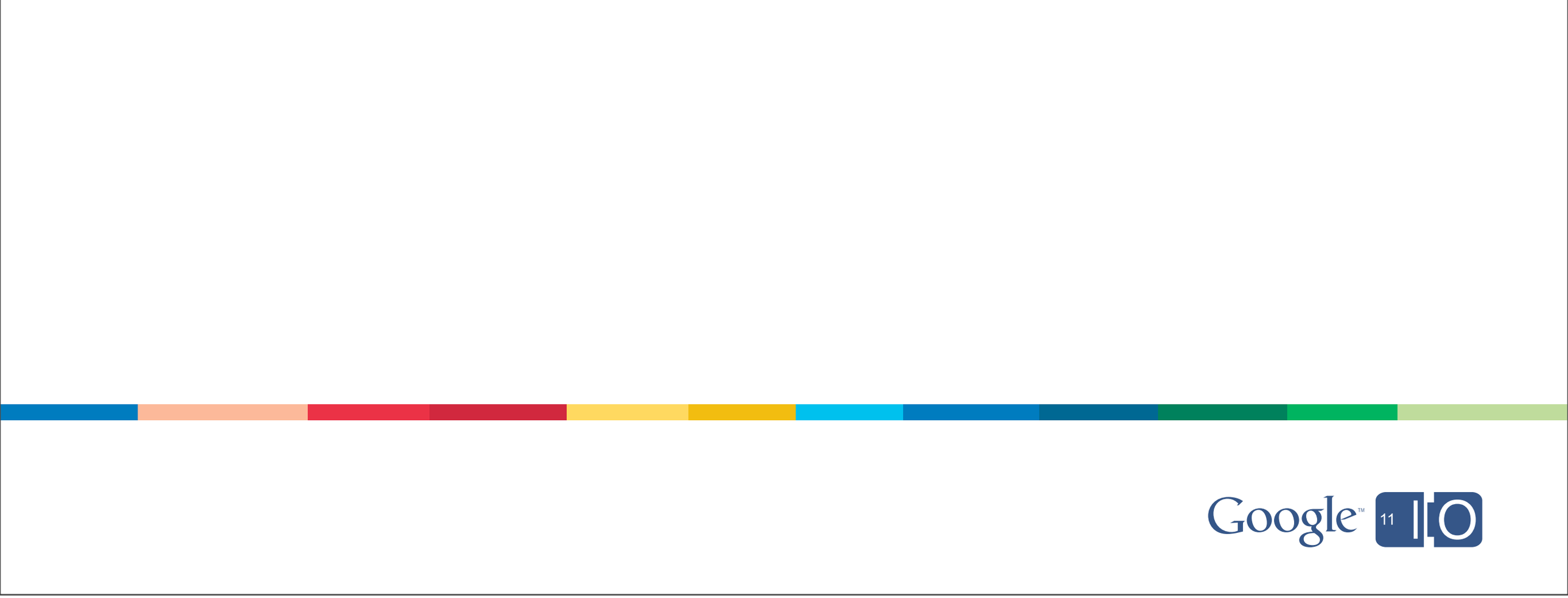
- Assists in the creation and development of Web Apps that use:
  - Google Web Toolkit (GWT)
  - App Engine for Java

- SDKs can be downloaded at the time of plugin install

- Design principle: blend naturally into Eclipse

Why are you here?

- You are...
  - an Eclipse user that's interested in GPE?
  - a GWT and/or App Engine user that's curious about GPE's new features?
  - interested in Google APIs and want to see how GPE can make it easier to use them?


Google™ 11

Getting your code from Google Project Hosting

Refresher: Google Project Hosting

- http://code.google.com/hosting
- Supports SVN and Mercurial repositories
- View repository and commit logs in your browser
  - can even make edits
- Issue tracking, wiki, project home page, ...
- No setup and maintenance headaches

(Demo - Google Project Hosting)

GPE and Project Hosting Integration

- Subclipse/Subversive/Mercurial handles version control functionality

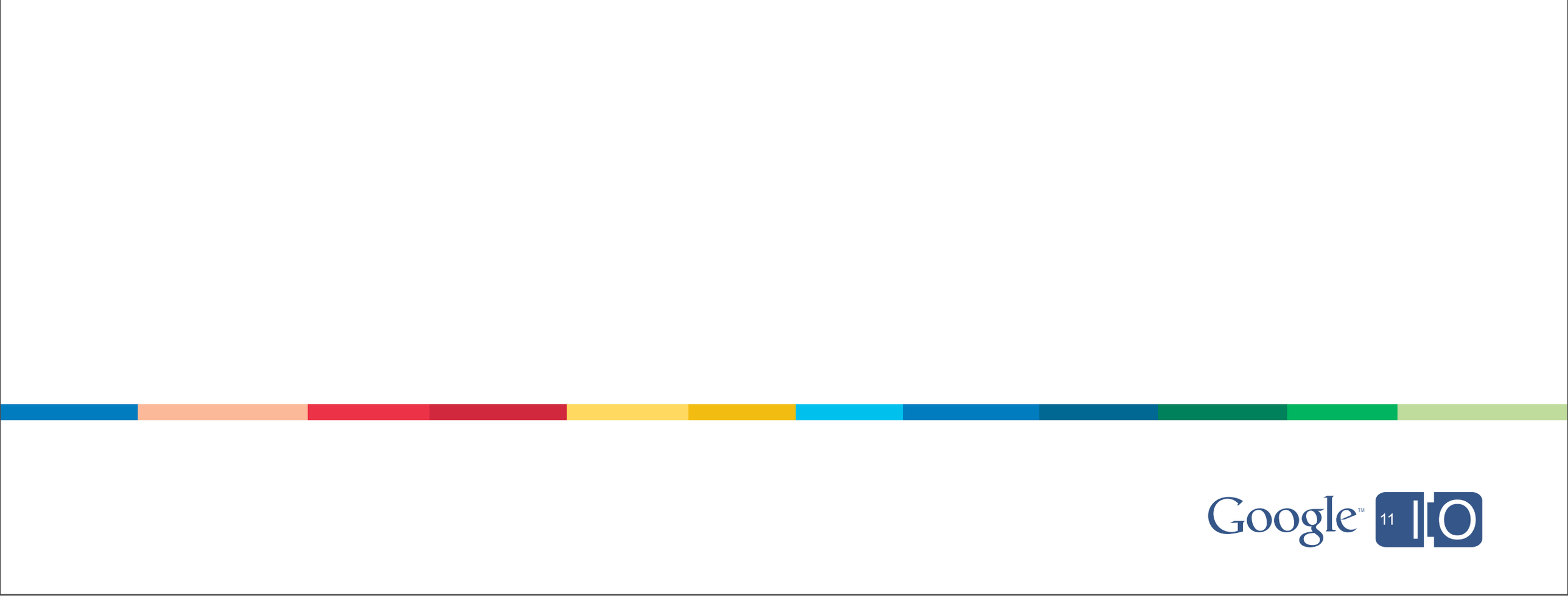
- Google Project Hosting API tells us:
  - project membership
  - Google code username/password

- Special installation of version control plugins
  - deferred installation = no up-front installation cost
  - detect conflicting version control providers

Storing GPE projects in Project Hosting

- When adding a new GPE project:
  - be sure to commit .project, .classpath, and .settings files
  - GPE needs this information to recognize your project

- Be careful: classpath container resolution depends on the SDKs that you have installed
  - Look at warnings: jars in war/WEB-INF/lib may be out of sync
  - Another option - get rid of containers, and add jars directly to your build path
  - Maven users don't have this problem


Google™ 11

Using Google APIs in your project

Google I/O 11

Google APIs

- Google has many APIs that provide access to services and user data
  - See them all: http://code.google.com/apis

- To name a few:
  - Maps, Earth, GMail, Docs, Buzz, YouTube

(Demo - Google APIs)

GPE and Google APIs

- GPE contacts a cloud service to get API list
  - API bundles are downloaded from this service

- APIs are added as ‘classpath containers’
  - API jars + dependency jars are added to build (and runtime) classpath
  - sources and javadoc are linked
  - jars are automatically copied to the WEB-INF/lib directory (configurable)

- Notifications for API updates appear next to the classpath container

GPE and Google APIs

- classpath container resolution is not an issue when storing Google API projects in version control
  - check in the .google_apis directory

- In v1, no clear distinction between server-side vs. client-side apis or build time vs. runtime dependencies
  - aggressive copies to WEB-INF/lib
  - everything ends up on the build path
  - should never end up with conflicting libraries, as long as upgrades are done in sync

- GWT Compilation will pick up GWT-based APIs

Developer Keys, Client Ids, and the API Console

- Need a developer key (or a client id) to use Google APIs
  - developer keys are used when user data is not involved

- Steps:
  - Create a project via the API console
    - http://code.google.com/apis/console
  - Enable the APIs that you are going to be using for the project
  - Make the developer key/client id accessible to the server-side of your app
    - for server-side apps, it should be secret (not publicly viewable)
    - for client-side apps, developer key is tied to serving site's domain

Accessing user information through the API

- If your app is trying to display Alice's Buzz posts...
  - authentication - Alice must prove her identity to Google
  - authorization - Alice must authorize access to her data
    - needs to tell Google that she's granted access for your app to access her Buzz information

- OAuth solves this problem
  - open standard implemented by Google, Facebook, Twitter, and others
  - different usage scenarios (native, web server, user agent)

- Google APIs that access user data all rely on OAuth
  - need a client id and client secret (dev key is not enough)

Want More Information on APIs an OAuth?

- "Identity and Data Access: OpenID and OAuth"
  - http://goo.gl/io/TmcV9

- "ClientLogin #FAIL"
  - http://goo.gl/io/HHQsq

- "Life of a Google API Developer"
  - http://goo.gl/io/aY5OD

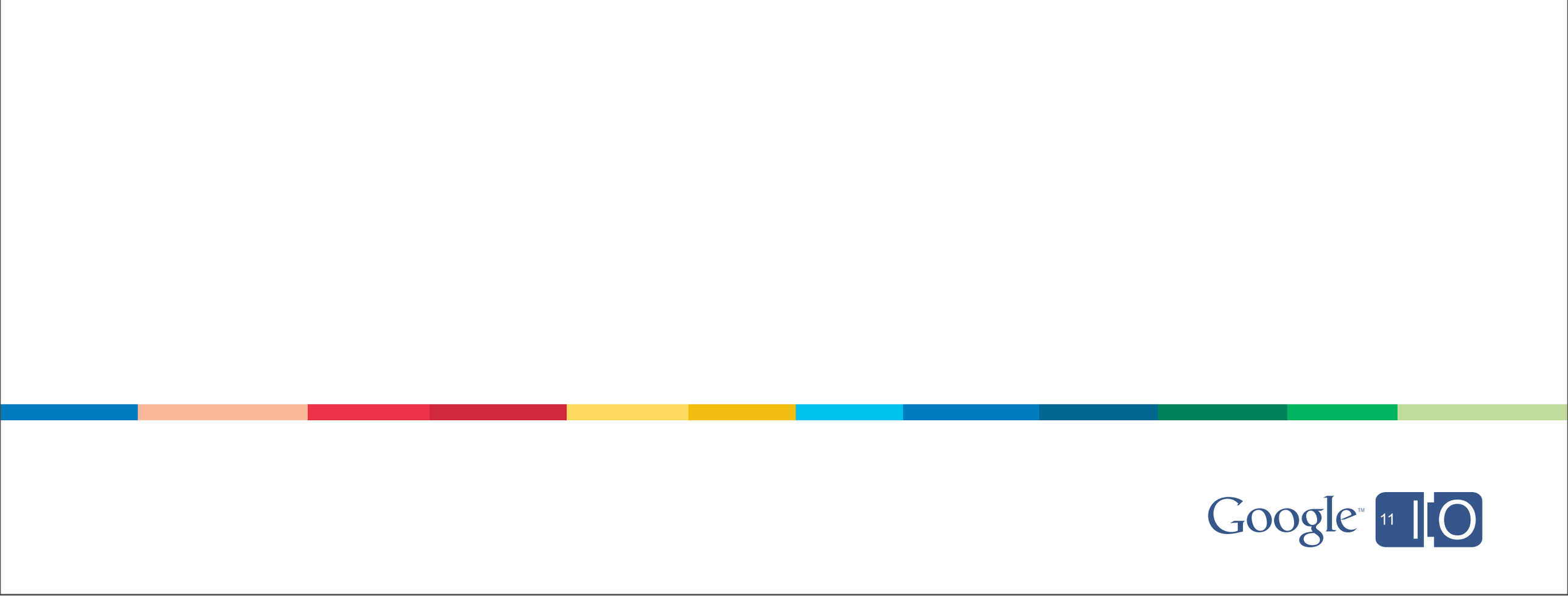
Rolling it all up..

- With a single click, see a list of Google's APIs
  - new ones automatically appear

- With one more click, you're ready to code against the API and access Google's services

- Push notifications that updates are available
  - rollbacks are possible if the update breaks your code

- Use the API console to enable APIs and get developer keys / client ids


Google™ 11

Deploying your project to Google App Engine

Refresher: What is App Engine?

- App Engine is a cloud computing platform
  - write a web app in Java (or Python) and deploy it
  - lives at <app id>.appspot.com

- Full set of services
  - hardware connectivity, JVM, servlet container, automatic scaling, datastore, distributed memory caching, task queueing,...

- security: your app is isolated from other running apps
  - sandbox - provides limited access to the underlying OS
  - certain JRE classes are off-limits

Deploying to Google App Engine

- In GPE, you can deploy an App Engine project with a single click

- App Engine 1.5.0 introduces the concept of resident backends
  - longer request deadlines; higher memory and CPU limits
  - in memory-state can be preserved across requests
  - individually addressable; can be manually started/stopped
  - Check out the “App Engine Backends” talk
    - http://www.google.com/events/io/2011/sessions/app-engine-backends.html

- Deploying to backends is now supported by GPE!

(Demo - Google App Engine)

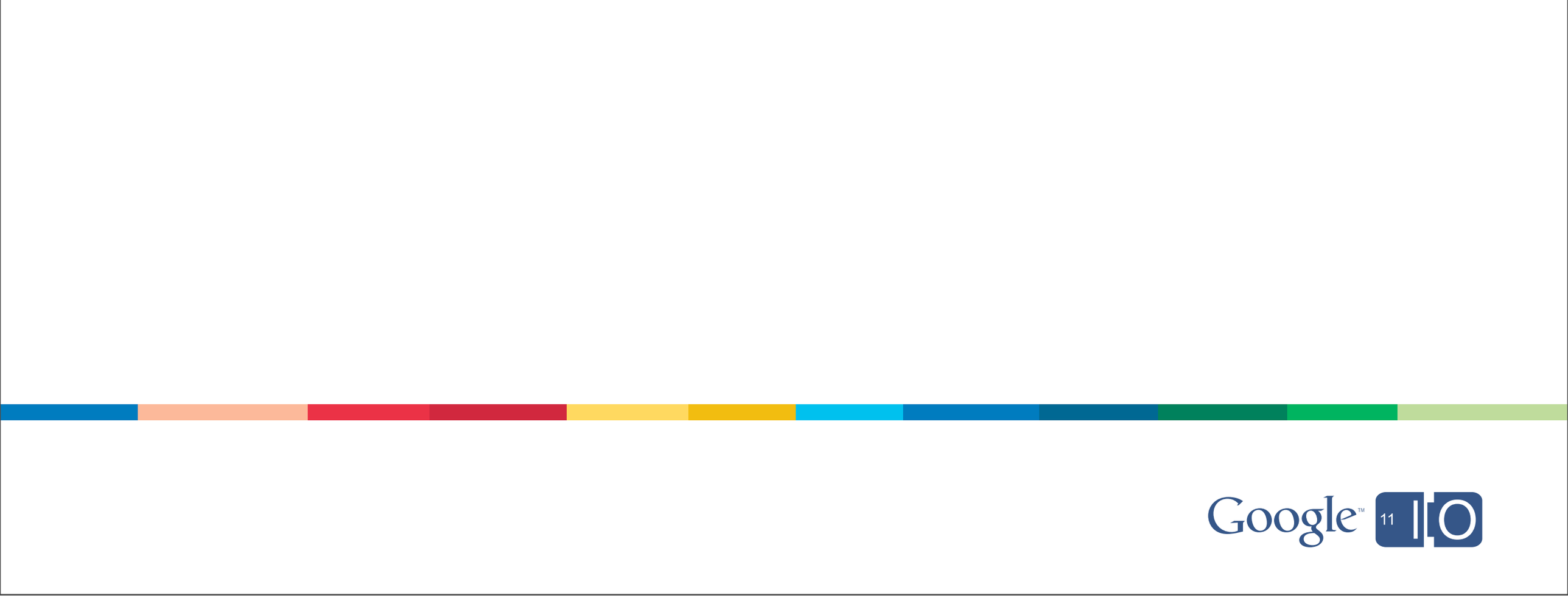
GPE and Google App Engine

- GPE uses an API provided by App Engine to upload the app
  - no application-specific passwords are necessary!
    - works for older versions of App Engine projects

- Presence of backends.xml indicates to GPE that there are different pieces to deploy
  - deployment occurs in sequence
  - deployment of backends automatically starts them up
  - Need to be using App Engine 1.5.0+

A few other notes..

- If your project uses GWT and App Engine, GWT compiles happen before deployment
  - a recompilation does not take place if there were no changes
- By default, ORM support is turned on
  - you can exclude all of your code from enhancement
  - or, disable the Enhancer builder, and remove the ORM-related jars from war/WEB-INF/lib
- GPE does not distinguish between server-side and client side-code
  - aggressive validations


Google™ 11

Quick Bits:
Other Cool Features

Single-Sign-On to Google's Services

- Using GPE, you're able to "log in" to Google's Services
  - specifically, App Engine and Project Hosting
  - if more are added, you'll be prompted to grant access
  - credentials persist between Eclipse sessions

- Achieved using OAuth
  - GPE does not have your username/password
  - OAuth token grants specific access to GPE for services that user has agreed to

- Logging out revokes access using the token
  - can also do this via your accounts page

GWT RPC Tooling in Eclipse

- GWT in 3 seconds
  - Develop and debug your web app in Java, cross-compile and it to JavaScript/HTML and deploy it
  - deals with the client-side, but has both high-level and low-level RPC mechanisms
    - traditional GWT RPC, RequestFactory

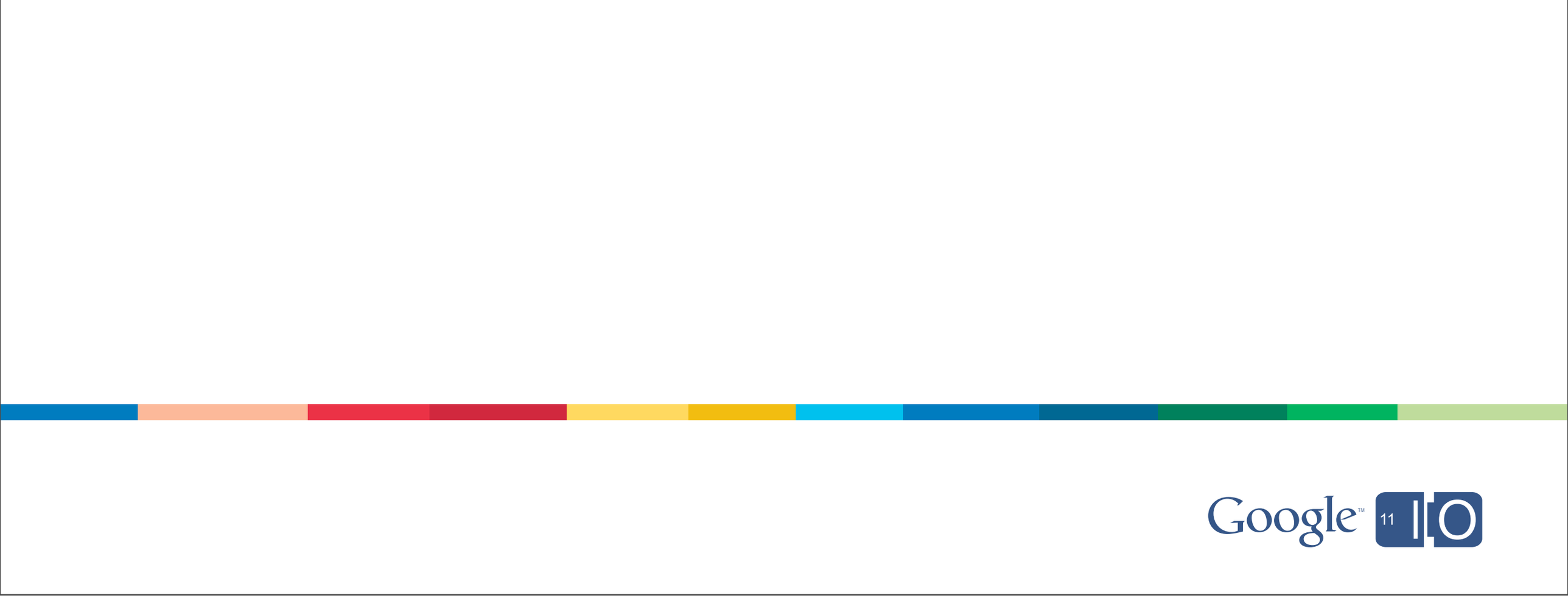
- RequestFactory
  - data-centric vs. service-centric
  - addresses the “DAO problem”
  - easy to derive a client data access layer from server layer
  - small payloads - only modified entities are sent over the wire

GWT RPC Tooling in Eclipse

- GPE makes it easy to use RequestFactory

- All of the required classes are generated based on an entity class
  - entity proxy
  - locator
  - RequestFactory interface
  - service stub
  - server-side skeleton implementation of service

- Check out the “Highly Productive GWT” talk
  - http://goo.gl/io/j8phu


Google™ 11

Wrap-up

Wrap-up

- GPE makes it easy for developers to leverage Google's cloud
  - Google Project Hosting, Google APIs, Google App Engine
- GPE's SSO functionality minimizes logins
  - allows users to manage their credentials
  - easy onramp for future Google service integration
- GPE 2.4 (beta) is a smooth onramp to powerful features in GWT and App Engine
  - RequestFactory
  - App Engine backends

Thank You

(Try GPE 2.4 Beta)

http://code.google.com/eclipse/beta/docs/download.html

(Session Feedback)

http://goo.gl/l7Z2y

(Hashtags)

#io2011 #DevTools

Resources

- Documentation
  - http://code.google.com/eclipse
- Update Sites (beta)
  - Eclipse 3.6 (Helios): http://dl.google.com/eclipse/plugin/beta/3.6
  - Eclipse 3.5 (Galileo): http://dl.google.com/eclipse/plugin//beta/3.5
  - Eclipse 3.4 (Ganymede): http://dl.google.com/eclipse/plugin/beta/3.4
- Google Web Toolkit and general GPE Questions
  - Group - http://groups.google.com/group/Google-Web-Toolkit
  - Issue Tracker - http://code.google.com/p/google-web-toolkit/issues
- App Engine Feedback
  - Group - http://groups.google.com/group/google-appengine-java
  - Issue Tracker - http://code.google.com/p/googleappengine/issues


Google™
11
I/O