

Google™



Storing Your Application's Data in the Google Cloud

Mike Schwartz & Navneet Joneja
May 10, 2011

Please provide feedback for this session at <http://goo.gl/oSZ7c>
Hashtags: #io2011, #AppEngine



Overview

- Introduction & What's New
- Getting Started
- More Advanced Features
- End to End Scenario

Introduction: What Is Google Storage?

- Store your data on Google's proven infrastructure
 - Reliable: data replicated to multiple data centers
 - Scalable, fast: Delivered via Google's worldwide network
- Developer / business focus
 - REST API
 - Flexible authentication and sharing
 - Supported by many 3rd party tools/libraries
 - Strong data consistency



Key Features (to Date)

Simple, Developer Oriented

- REST API
- Many SDKs and tools

Reliable, Scalable & Fast

- Any amount of data, up to 1 TB/Object
- Resumable transfers
- Replication to multiple US data centers

Control & Share your Data

- Signature-based authentication
- User & group ACLs
- Authenticated & anonymous browser downloads

New!



Simple, Developer Oriented

- REST API
- Many SDKs and tools
- Team-based account model
- Management through the API Console

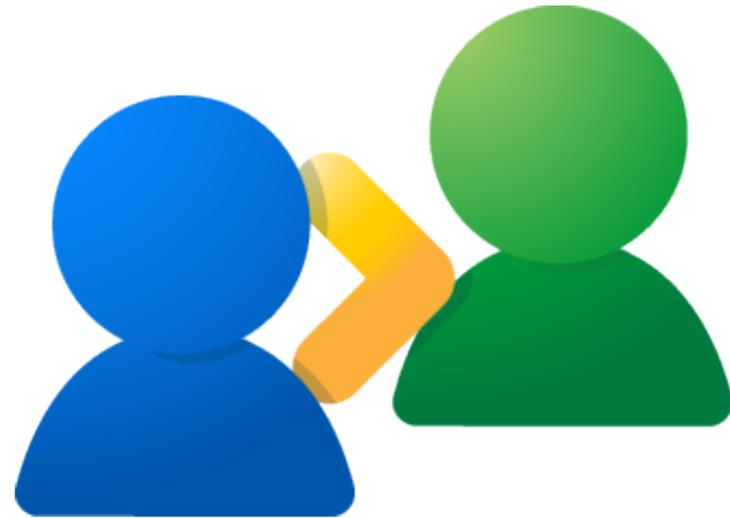
New!



Reliable, Scalable & Fast

- Any amount of data, Up to **5 TB/object**
- Resumeable transfers
- Replicated to multiple US or **European** data centers
- **Streaming uploads**
- **High performance that just works**

New!



Control & Share your Data

- Signature-based authentication
- User & group ACLs
- Authenticated & anonymous browser downloads
- OAuth 2.0
- Project team-oriented ACLs
- Share with anyone (with a Google account)

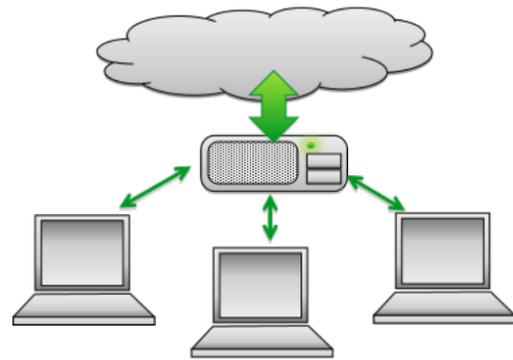
New!

Generally Available

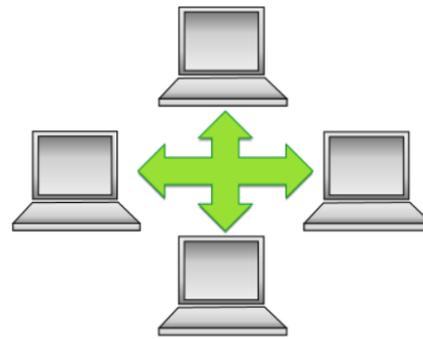
Free trial use through 12/31/2011

- 5 GB Storage
- 25 GB transfer into Google (20 GB US/EU, 5 GB Asia)
- 25 GB transfer out of Google (20 GB US/EU, 5 GB Asia)
- 2,500 Write requests
- 25,000 Read requests
- (first project)

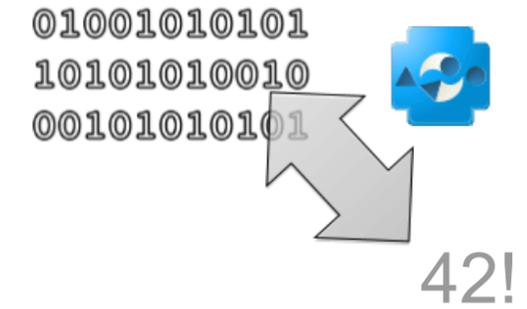
Example Use Cases



Storage for applications



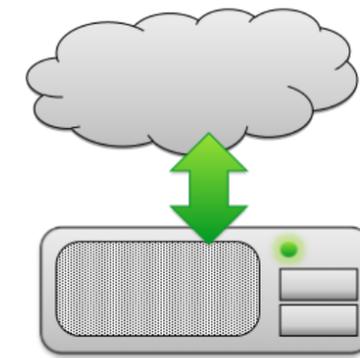
Data Sharing



Storage for computation

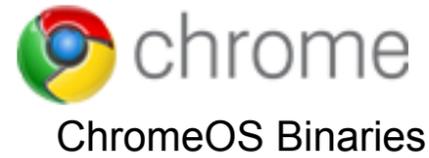


Static content hosting



Backup and recovery

Google Products using Google Storage



Introduction / What's New
Getting Started
More Advanced Features
End to End Scenario

Getting Started in < 1 Minute

- Sign Up: <http://gsdemo2011.com/get-gs>
- Get your data in/out using gsutil

Basic API

- Data naming: Buckets and objects
- RESTful API (HTTP verbs and URIs)

```
PUT /data.csv HTTP/1.1  
Host: gsdemo2011.commondatastorage.googleapis.com  
Authorization: OAUTH ...
```

- Additional functionality via HTTP headers

```
Content-Type: text/xml  
Content-Encoding: gzip  
Cache-Control: public,max-age=3600
```

Reference Implementations



GSUtil tool

```
gsutil -d cat gs://gsdemo2011/data.csv
```



boto library

```
from boto import storage_uri
uri = storage_uri('gs://gsdemo2011/data.csv')
print uri.get_contents_as_string()
```



REST API

```
GET /data.csv HTTP/1.1
Host: gsdemo2011.commondatastorage.googleapis.com
Authorization: OAUTH ...
```

Security and Sharing

- Three ways to authenticate requests
 - OAuth2
 - Web browser login
 - HMAC request signing

```
PUT /obj.csv HTTP/1.1
```

```
Host: gsdemo2011.com.commondatastorage.googleapis.com
```

```
Content-Length: 2190
```

```
Authorization: OAuth 1/83K32fmU8Uit392w01v3YR03jDi31j9-ijV1J341I10
```

Security and Sharing (continued)

- Buckets and objects have ACLs
- GET/PUT URI?acl

```
<AccessControlList>
  <Owner><ID>...</ID></Owner>
  <Entries>
    <Entry>
      <Scope type="UserByEmail">
        <EmailAddress>gsdemo2011@gmail.com</EmailAddress>
      </Scope>
      <Permission>FULL_CONTROL</Permission>
    </Entry>
    <Entry>
      <Scope type="GroupByEmail">
        <EmailAddress>gsdemo2011-guests@googlegroups.com</EmailAddress>
      </Scope>
      <Permission>READ</Permission>
    </Entry>
  </Entries>
</AccessControlList>
```

Introduction / What's New
Getting Started

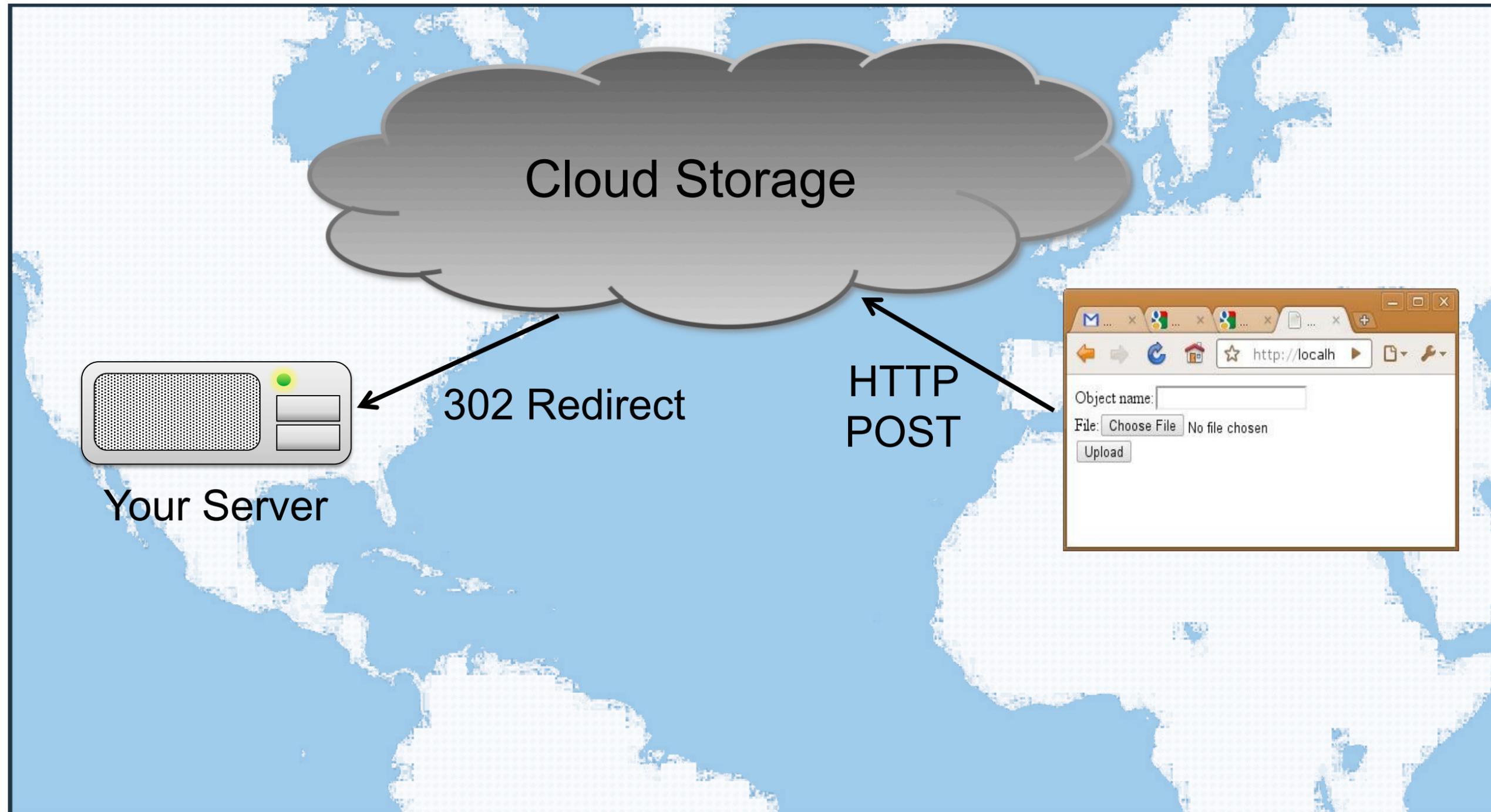
More Advanced Features

End to End Scenario

European Buckets

- Optional location specified at bucket creation
- Location lookup
 - `GET bucket/?location`
- Strong consistency
 - Once bucket created it can immediately be used from anywhere in the world

Strong Data Consistency



Resumable Transfers

- Resumable uploads use Google Data Resumable protocol

```
PUT /gsdemo2011.commondatastorage.googleapis.com/?upload_id=AEnB2...
```

```
HTTP/1.1 308 Resume Incomplete  
[header: Range: bytes=0-1048575]
```

```
PUT /gsdemo2011.commondatastorage.googleapis.com/?upload_id=AEnB2  
Content-Range: bytes 1048576-2276223/2276224  
...
```

- Resumable downloads use range GETs

```
GET /gsdemo2011.commondatastorage.googleapis.com  
[header: Range: bytes=7806976-10485759]  
...
```

OAuth2

- Widely adopted open standard
- Authorize web apps without sharing login/password
- Authorize different apps with separate tokens
- No request signing needed

Introduction / What's New
Getting Started
More Advanced Features

End to End Scenario

Scenario: Photographers' Preview Service

- Web app enabling upload/watermarking, and sale of original photos
- Demonstrates:
 - Team-based development & management
 - Domain-named buckets
 - OAuth 2.0 authorization
 - Group sharing

Scenario: Photographers' Preview Service



- Create project
- Register App
- Verify DNS owner
- Create Bucket

Scenario: Photographers' Preview Service

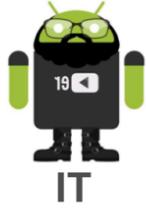


- Create project
- Register App
- Verify DNS Owner
- Create Bucket



- Implement app
- Test (project-private)
- Deploy

Scenario: Photographers' Preview Service



IT

- Create project
- Register App
- Verify DNS Owner
- Create Bucket / ACL



Developers

- Implement App
- Test (project-private)
- Deploy



- Upload photos

- 1 • Run app
- 3 • Authorize access

- 2 • Request credentials
 - Read & watermark photos
 - Handle orders

Scenario: Photographers' Preview Service



IT

- Create project
- Register App
- Verify DNS Owner
- Create Bucket / ACL



Developers

- Implement App
- Test (project-private)
- Deploy



Photographer

- Upload photos
- Run App
- Authorize Access



App

- Request Credentials
- Read & Watermark Photos
- Handle orders



- Join group
- View and order photos

Scenario: Photographers' Preview Service



- Create project
- Register App
- Verify DNS owner
- Create bucket



- Implement App
- Test (project-private)
- Deploy



- Upload photos
- Run App
- Authorize access

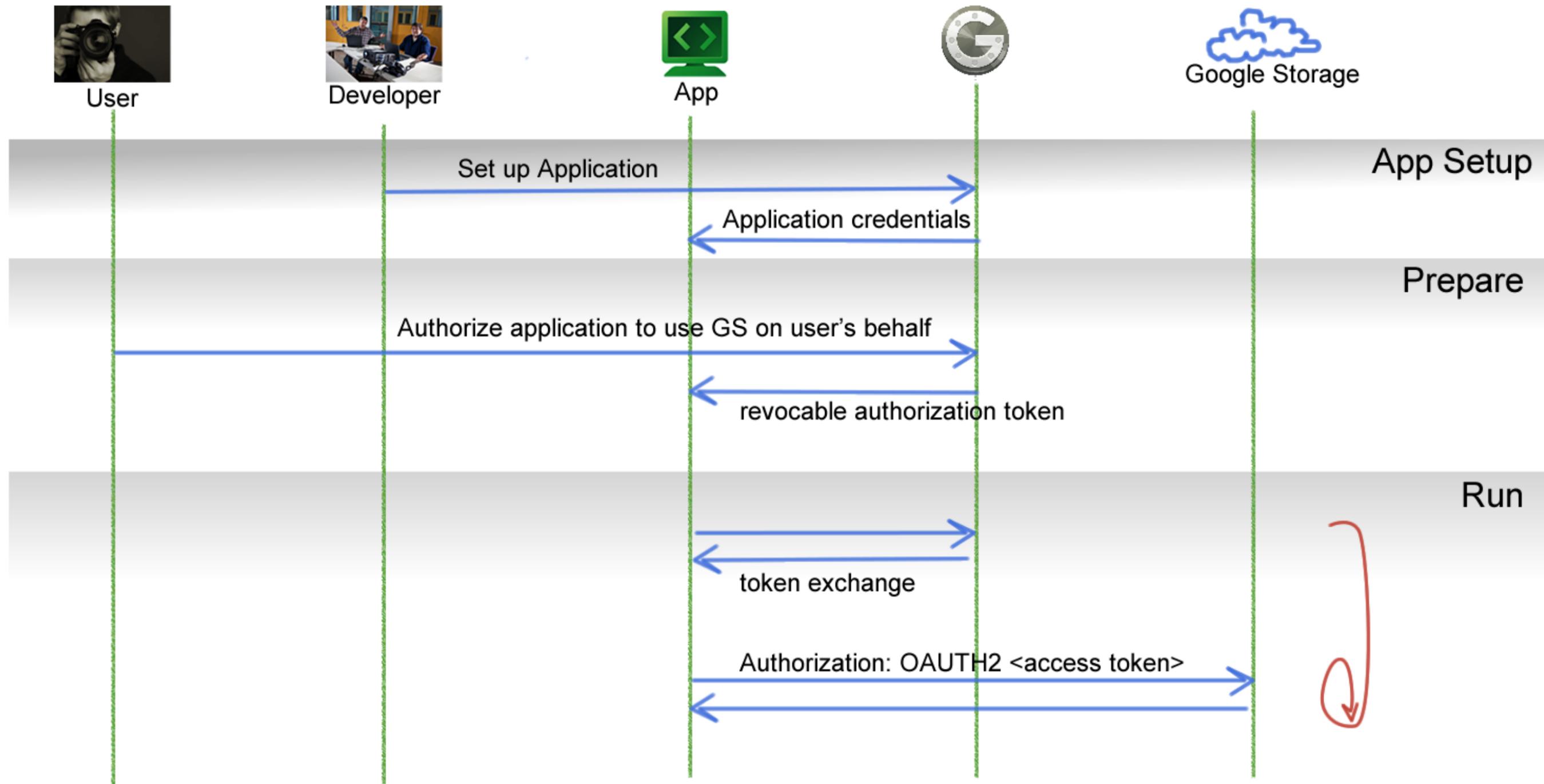


- Request credentials
- Read & watermark photos
- Handle orders



- Join group
- View and order photos

OAuth 2.0 Three-Legged Authorization [see gsutil/cloudauth]



Recap: Developer / Business features

- Transport support for streaming and large transfers
- Specify geo location: US (default) or Europe
- Projects, team based development & management, API console
- Ownership verification for domain-named buckets
- Security:
 - OAuth 2.0
 - Group sharing

Get Google Storage for Developers now!

<http://gsdemo2011.com/get-gs>

Session feedback: <http://goo.gl/oSZ7c>

Google™



Additional Detail Slides

Excerpted OAuth 2 Webapp

```
class MainHandler(webapp.RequestHandler):
    def get(self):
        credentials = StorageByKeyName(...).get()
        if not credentials or credentials.invalid:
            flow = OAuth2WebServerFlow(client_id, client_secret,
                scope, user_agent, auth_uri, token_uri)
            callback = self.request.relative_url('/auth')
            auth_url = flow.step1_get_authorize_url(callback)
            memcache.set(user.user_id(), pickle.dumps(flow))
            self.redirect(auth_url)
        else:
            self.response.out.write(uri.get_contents_as_string())

class OAuthHandler(webapp.RequestHandler):
    def get(self):
        flow = pickle.loads(memcache.get(user_id))
        credentials = flow.step2_exchange(self.request.params)
        StorageByKeyName(...).put(credentials)
        self.redirect("/")

def main():
    run_wsgi_app(webapp.WSGIApplication(
        [('/', MainHandler), ('/auth', OAuthHandler)]))
```