How Google Builds APIs

Zach Maier, Mark Stahl,
Joey Schorr, Yaniv Inbar

May 19-20, 2010

How Google Builds APIs

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

http://bit.ly/apiwave
#googleapis8

How Google Builds APIs

How Google Builds APIs

- Google APIs 101

How Google Builds APIs

- Google APIs 101
- Making Future APIs Awesome

How Google Builds APIs

- Google APIs 101
- Making Future APIs Awesome
- How Google *Really* Builds APIs

How Google Builds APIs

- Google APIs 101
- Making Future APIs Awesome
- How Google *Really* Builds APIs
- Questions and Comments

How Google Builds APIs

- **Google APIs 101**
- Making Future APIs Awesome
- How Google *Really* Builds APIs
- Questions and Comments

REST 101

- REST == Representational State Transfer
- Client and servers transferring resource representations
- Good for cached and layered systems (like the web)

REST 101

- REST == Representational State Transfer
- Client and servers transferring resource representations
- Good for cached and layered systems (like the web)

- In HTTP, this means verbs acting on resource URIs

  GET http://gdata.youtube.com/feeds/api/channels

REST 101

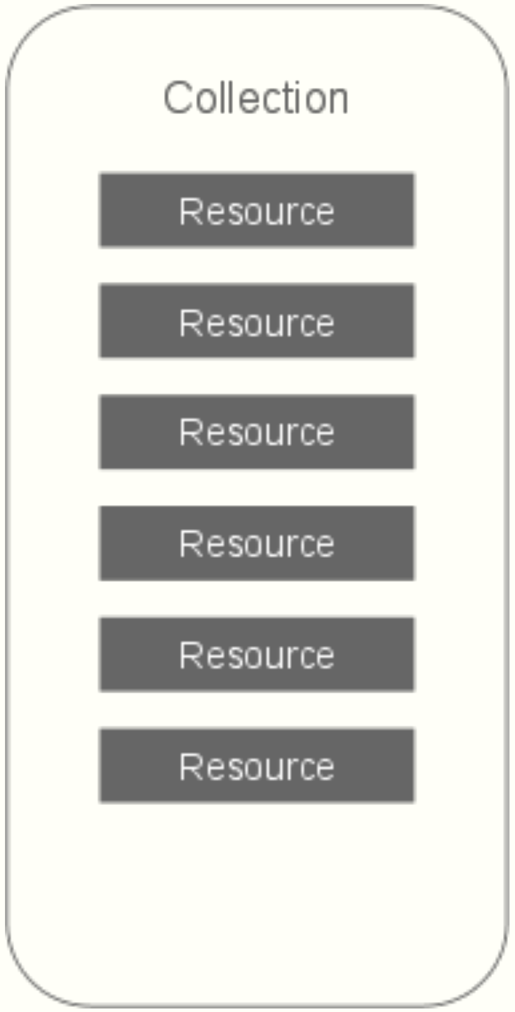
- REST == Representational State Transfer
- Client and servers transferring resource representations
- Good for cached and layered systems (like the web)

- In HTTP, this means verbs acting on resource URIs

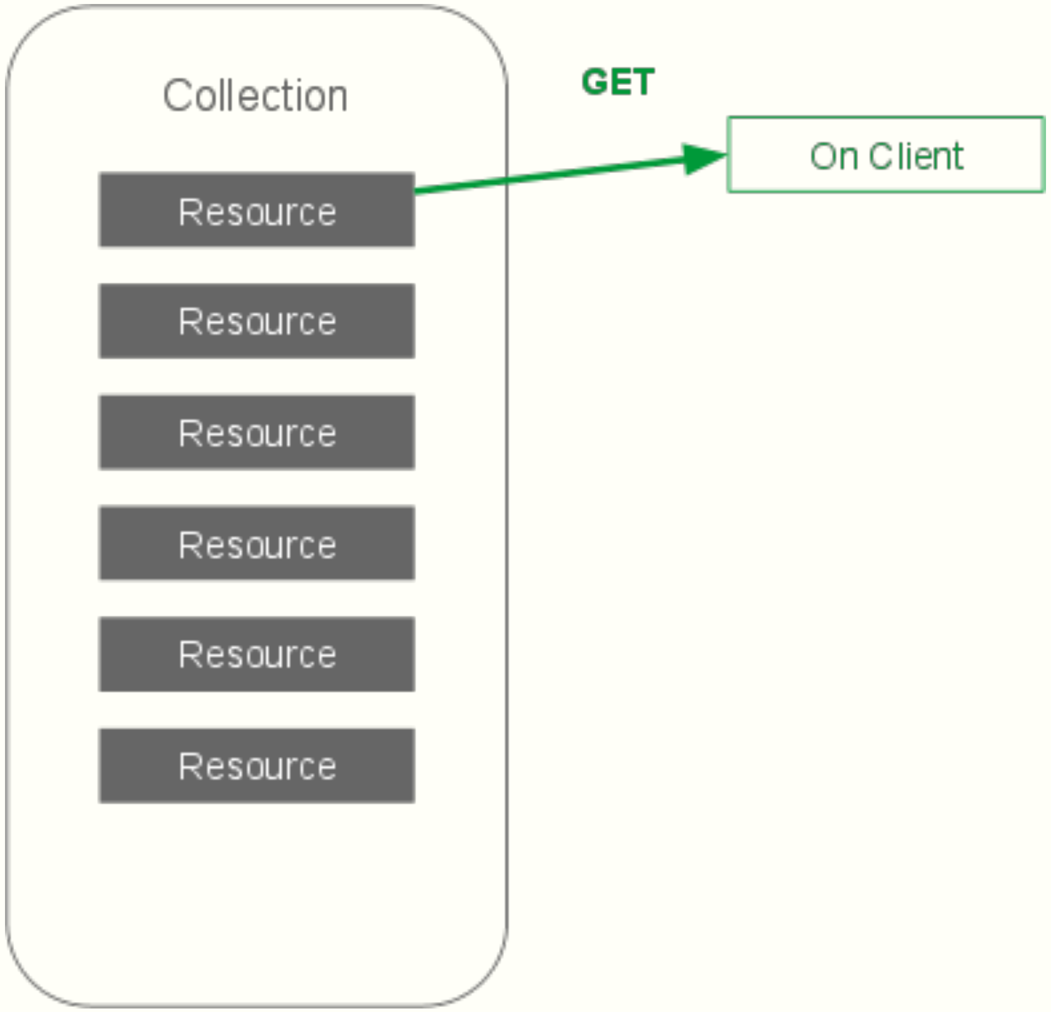
GET http://gdata.youtube.com/feeds/api/channels

- AtomPub models data as feeds of entries
- Or more generally, as collections of resources

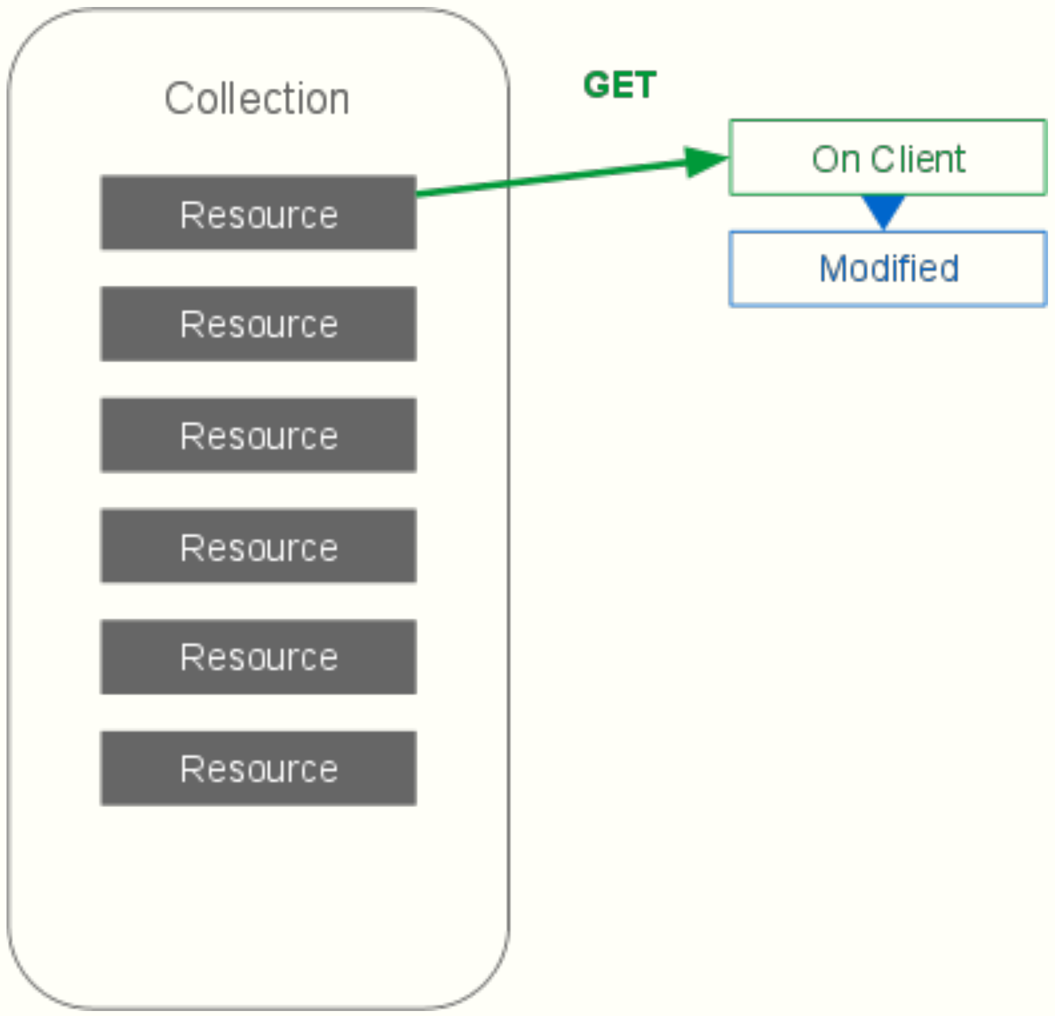
REST 101


Collection
Resource
Resource
Resource
Resource
Resource
Resource

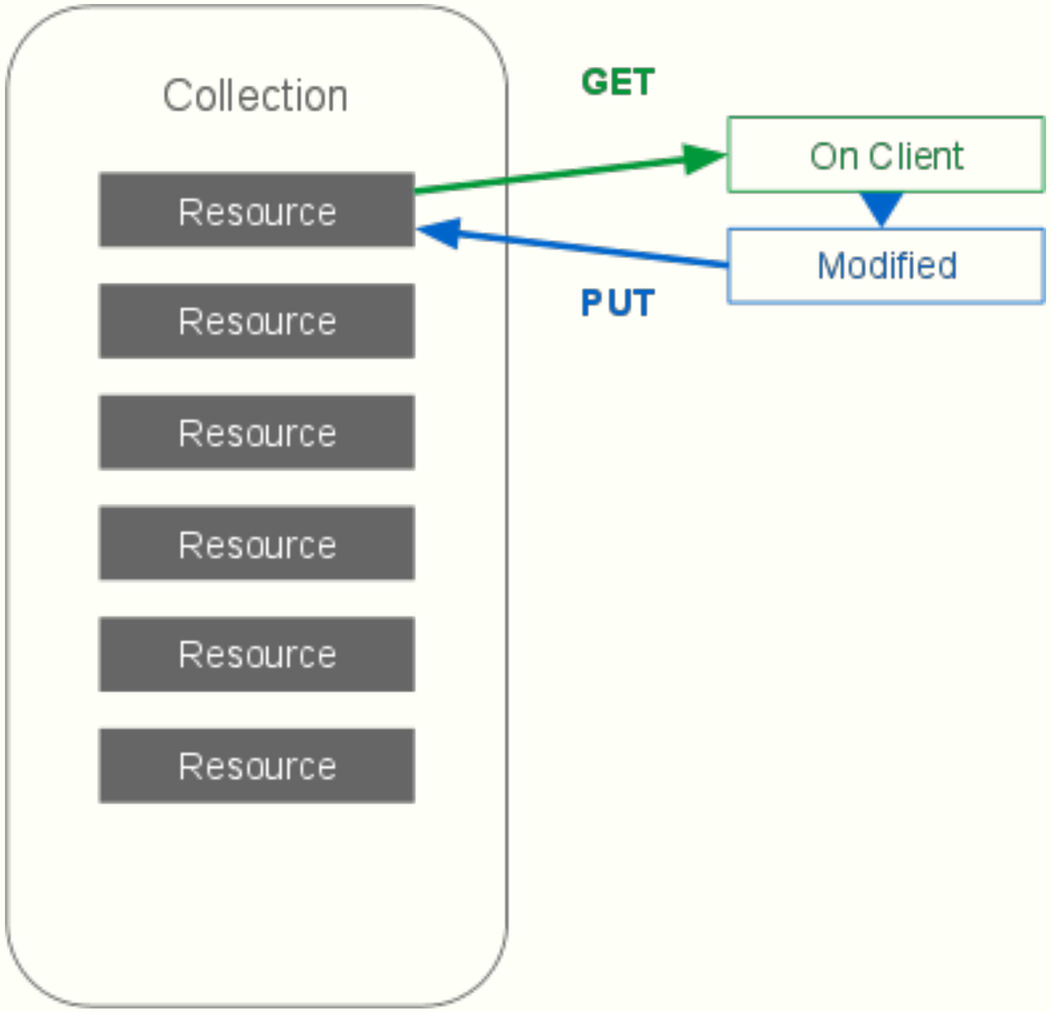
REST 101


Collection
Resource
Resource
Resource
Resource
Resource
Resource
GET
On Client

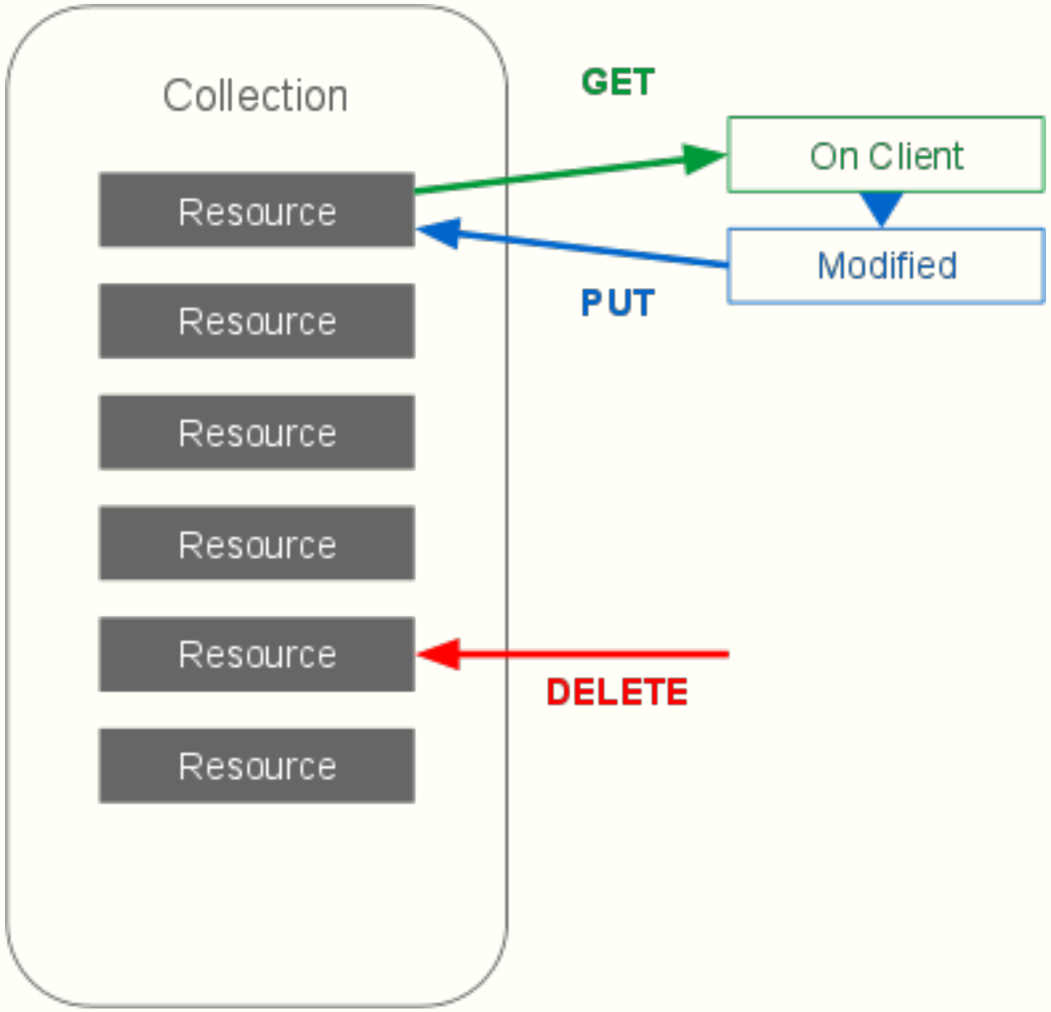
REST 101


Collection
Resource
Resource
Resource
Resource
Resource
Resource
GET
On Client
Modified

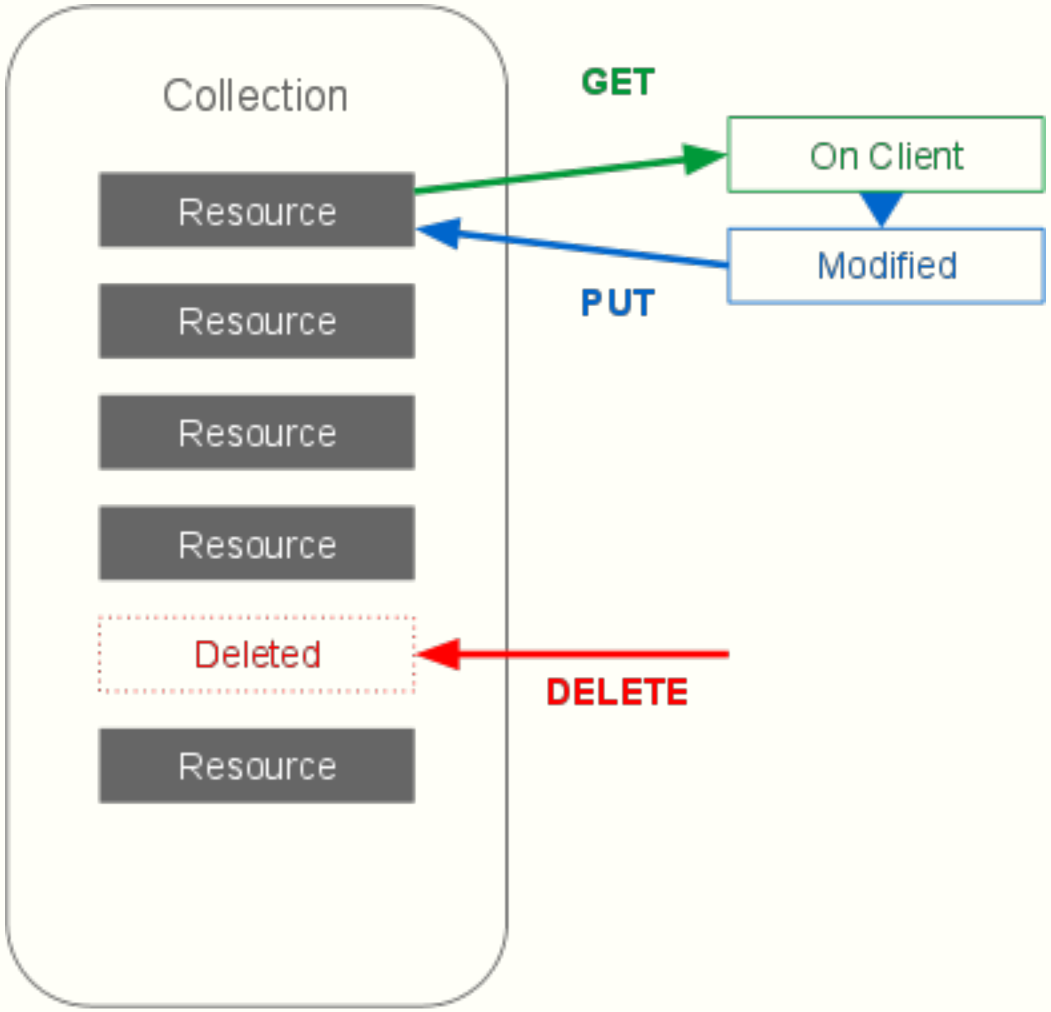
REST 101


Collection
Resource
Resource
Resource
Resource
Resource
Resource
GET
On Client
Modified
PUT

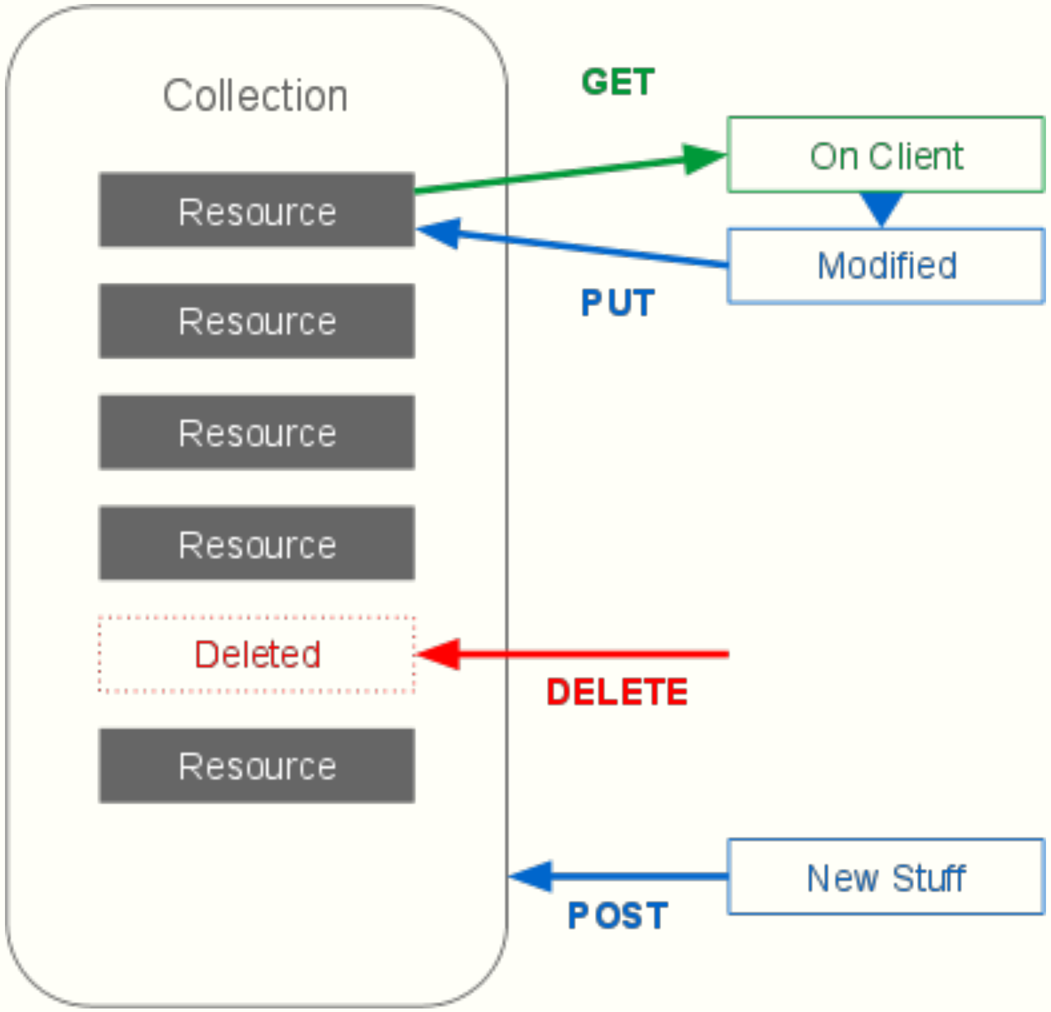
REST 101


Collection
Resource
Resource
Resource
Resource
Resource
Resource
GET
On Client
Modified
PUT
DELETE

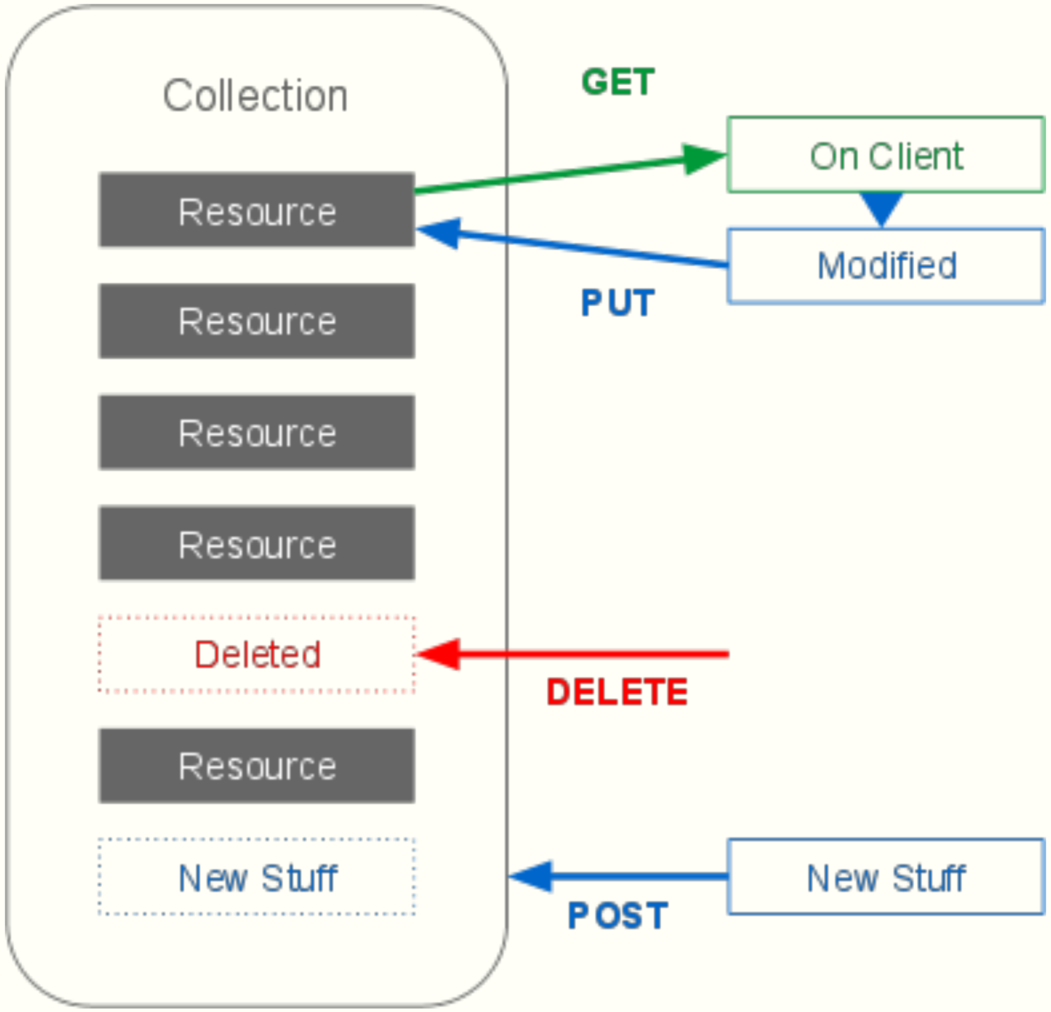
REST 101


Collection
Resource
Resource
Resource
Resource
Deleted
Resource
GET
On Client
Modified
PUT
DELETE

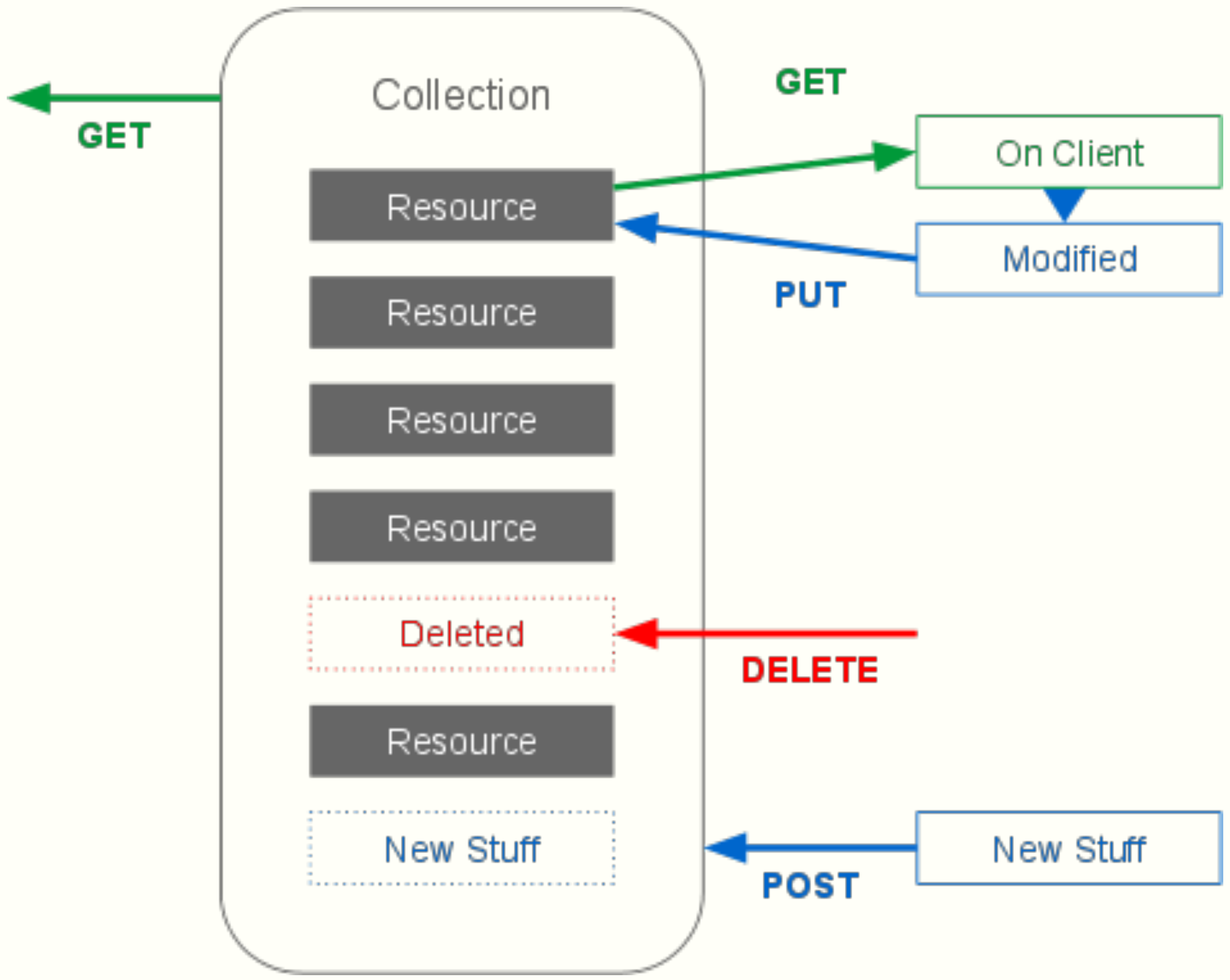
REST 101


Collection
Resource
Resource
Resource
Resource
Deleted
Resource
GET
On Client
Modified
PUT
DELETE
New Stuff
POST

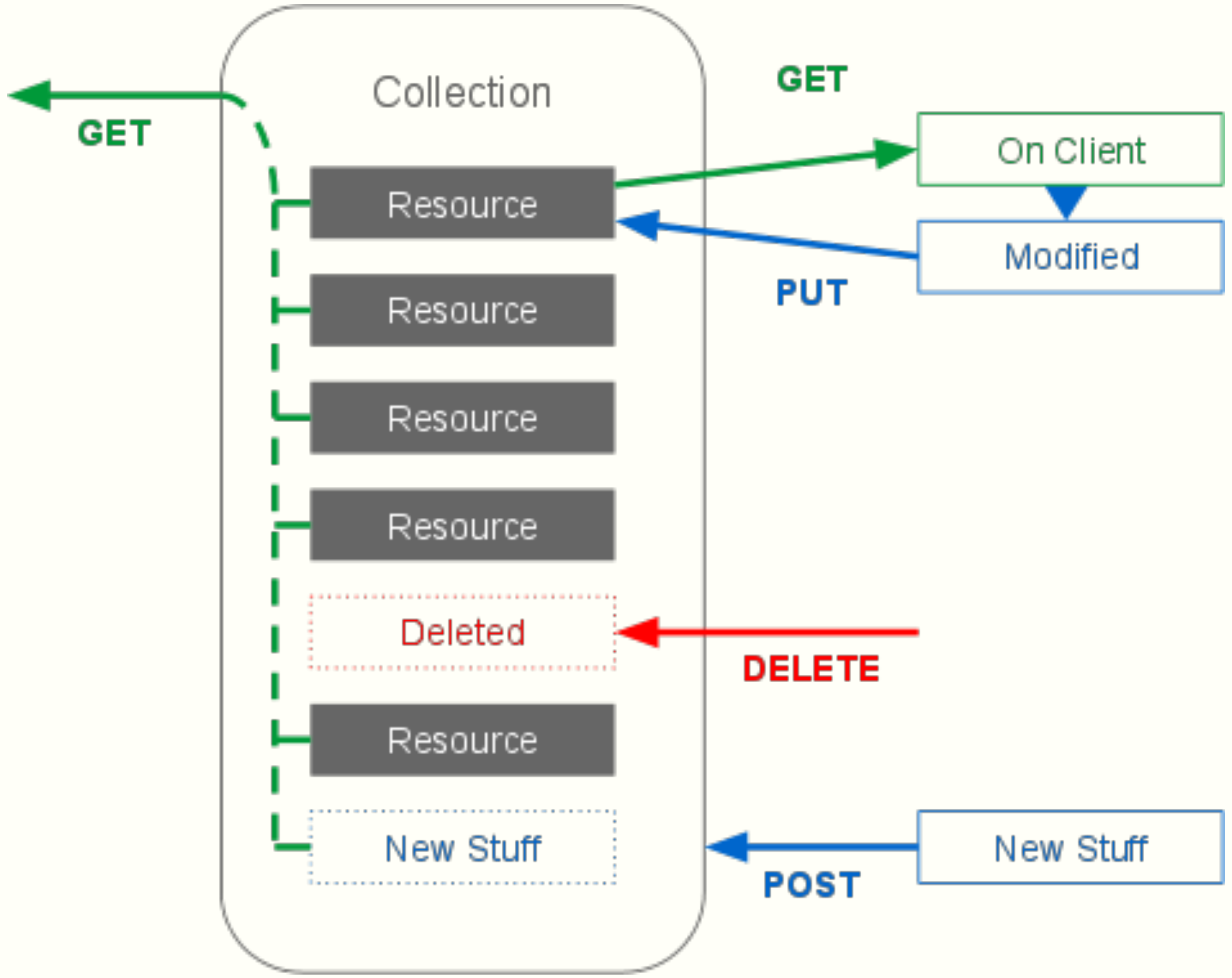
REST 101


Collection
Resource
Resource
Resource
Resource
Deleted
Resource
New Stuff
GET
On Client
Modified
PUT
DELETE
New Stuff
POST

REST 101


GET
Collection
Resource
Resource
Resource
Resource
Deleted
Resource
New Stuff
GET
On Client
Modified
PUT
DELETE
New Stuff
POST

REST 101


Collection
GET
GET
Resource
On Client
Modified
PUT
Resource
Resource
Resource
Deleted
DELETE
Resource
New Stuff
POST
New Stuff

Google Data

Google Data

Google Data ≈ Atom

- Right now you MUST understand Atom to use the APIs.

Google Data ≈ Atom

- Right now you MUST understand Atom to use the APIs
- **Core built around...**
  - Atom Syndication Format (RFC4287)
  - Atom Publishing Protocol (RFC5023)

Google Data ≈ Atom

- Right now you MUST understand Atom to use the APIs

- Core built around...
  - Atom Syndication Format (RFC4287)
  - Atom Publishing Protocol (RFC5023)

- **Extended the core features**
  - Query parameters
  - Concurrency
  - Batch

Google Data

- More than 25 APIs
- More than 2B hits per day across all APIs.

The Future Google APIs

- **We're moving to a brand new API infrastructure**

The Future Google APIs

- **We're moving to a brand new API infrastructure**

- You might already be using it...

Google moderator

Google buzz

Google latitude

How Google Builds APIs

- Google APIs 101
- **Making Future APIs Awesome**
- How Google *Really* Builds APIs
- Questions and Comments

The Future Google APIs

- Rough edges in Google APIs
    - Output formats
    - Calling styles
    - Client libraries

Output Formats

Output Formats

- **Issue: Resources can be verbose**

Output Formats

- **Solution: Allow operations on partial data**

Output Formats

- **Solution: Allow operations on partial data**
- Partial Response: **GET** just the fields you want

Output Formats

- **Solution: Allow operations on partial data**
- Partial Response: **GET** just the fields you want

```
http://gdata.youtube.com/feeds/api/videos
    ?v=2&q=google%20io&max-results=1
    &alt=xml
```

bit.ly/nopartial

Output Formats

- **Solution: Allow operations on partial data**
- Partial Response: **GET** just the fields you want

```
http://gdata.youtube.com/feeds/api/videos
    ?v=2&q=google%20io&max-results=1
    &alt=xml
```

bit.ly/nopartial

```
http://gdata.youtube.com/feeds/api/videos
    ?v=2&q=google%20io&max-results=1
    &alt=xml&fields=entry(title,content)
```

bit.ly/partialon

Output Formats

- **Solution: Allow operations on partial data**
- Partial Response: **GET** just the fields you want

  http://gdata.youtube.com/feeds/api/videos
  ?v=2&q=google%20io&max-results=1
  &alt=xml
  bit.ly/nopartial

  http://gdata.youtube.com/feeds/api/videos
  ?v=2&q=google%20io&max-results=1
  &alt=xml&fields=entry(title,content)
  bit.ly/partialon

- Partial Update: **PATCH** just the fields you retrieved

Output Formats

- **Solution: Allow operations on partial data**
- Partial Response: **GET** just the fields you want

  http://gdata.youtube.com/feeds/api/videos
  ?v=2&q=google%20io&max-results=1
  &alt=xml
  bit.ly/nopartial

  http://gdata.youtube.com/feeds/api/videos
  ?v=2&q=google%20io&max-results=1
  &alt=xml&fields=entry(title,content)
  bit.ly/partialon
- Partial Update: **PATCH** just the fields you retrieved
- More documentation at http://bit.ly/partialops

Output Formats

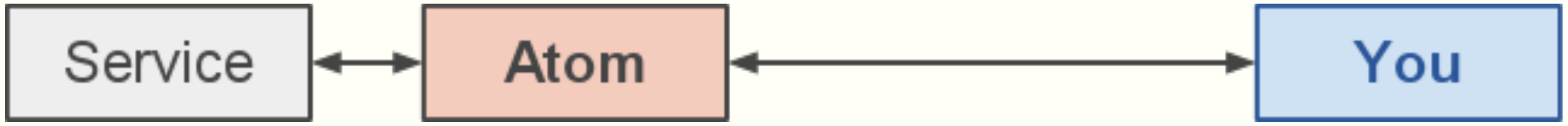
- **Issue: XML isn't easy on all platforms**

Output Formats

- **Solution: Support multiple read-write formats**

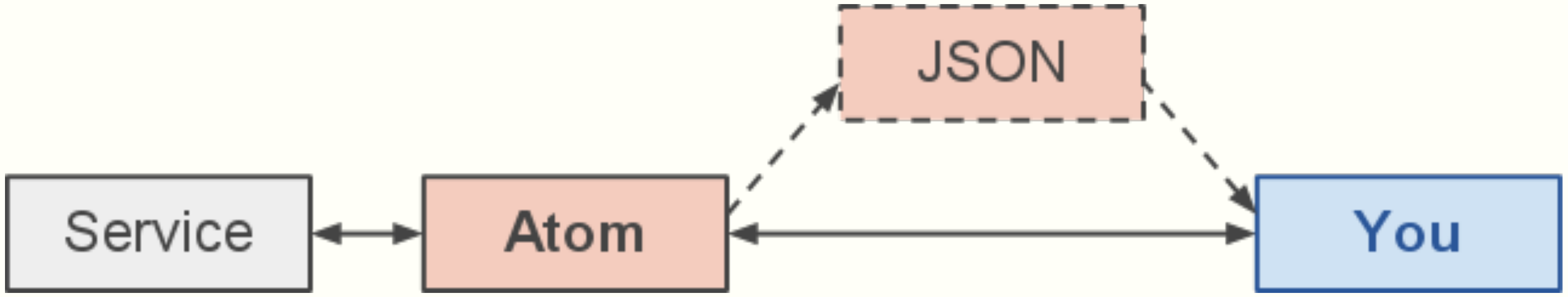
Output Formats

- Solution: Support multiple read-write formats
    - Architecture Issue: Protocol-specific data model



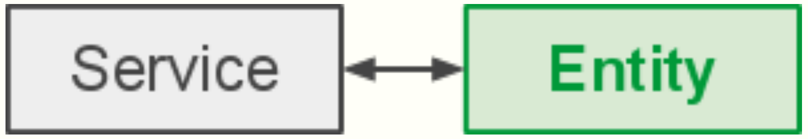
Output Formats

- Solution: Support multiple read-write formats
  - Architecture Issue: Protocol-specific data model



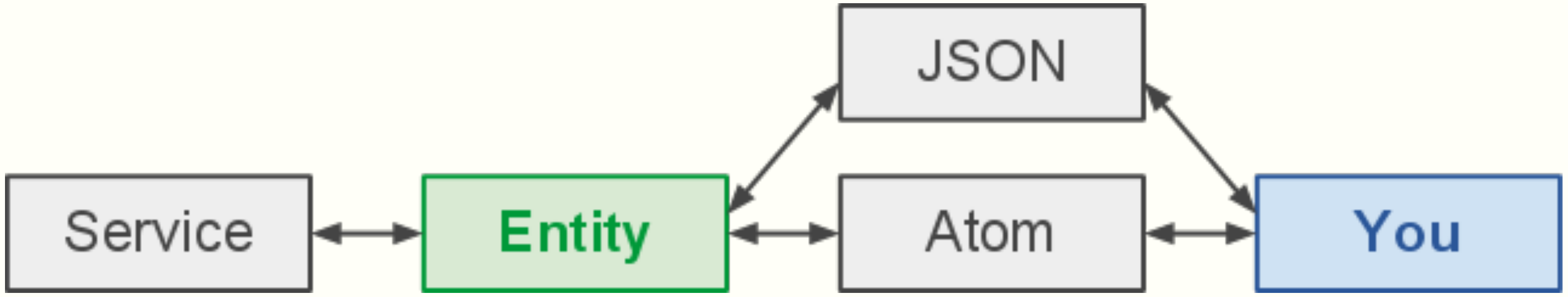
Output Formats

- **Solution: Support multiple read-write formats**
  - **Architecture Solution: Generic data model**



Output Formats

- **Solution: Support multiple read-write formats**
    - **Architecture Solution: Generic data model**



Output Formats

- **Solution: Support multiple read-write formats**
    - **Architecture Solution: Generic data model**

http://www.googleapis.com/buzz/v1/activities
/markstahl/@public?q=google&max-results=1
&**alt=atom**

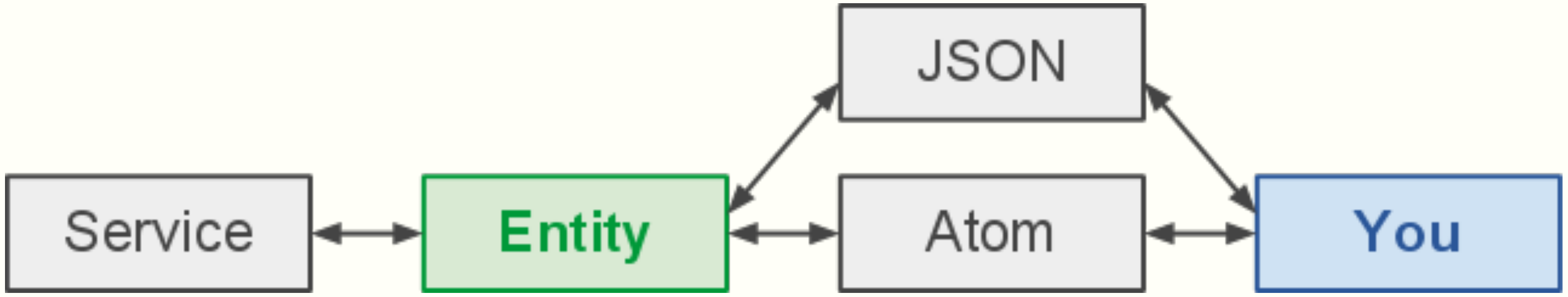
bit.ly/buzzatom

http://www.googleapis.com/buzz/v1/activities
/markstahl/@public?q=google&max-results=1
&**alt=json**

bit.ly/buzzjson

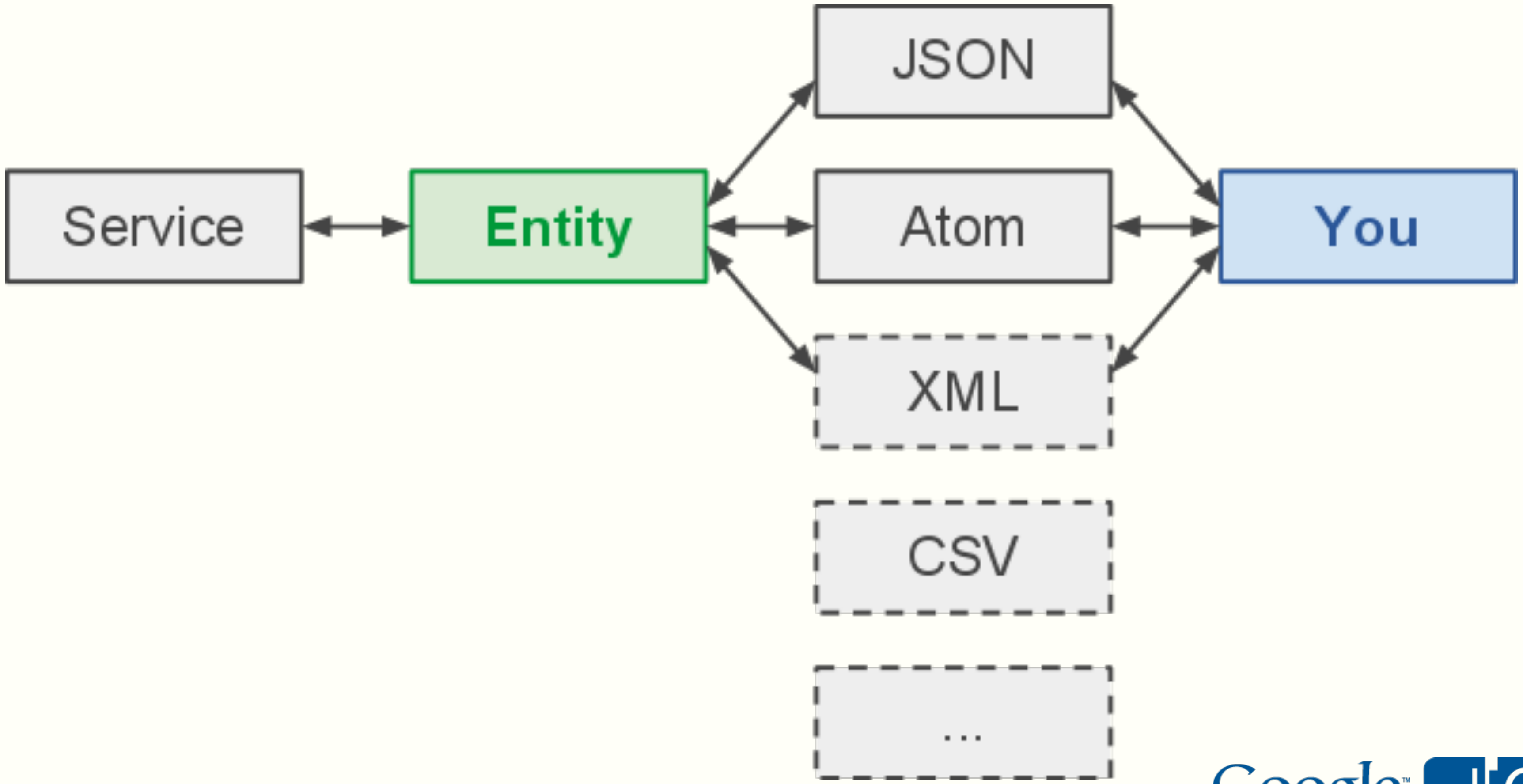
Output Formats

- **Solution: Support multiple read-write formats**
  - **Architecture Solution: Generic data model**



Output Formats

- **Solution: Support multiple read-write formats**
  - **Architecture Solution: Generic data model**



Calling Styles

Calling Styles

- **Issue: REST can be Awkward**

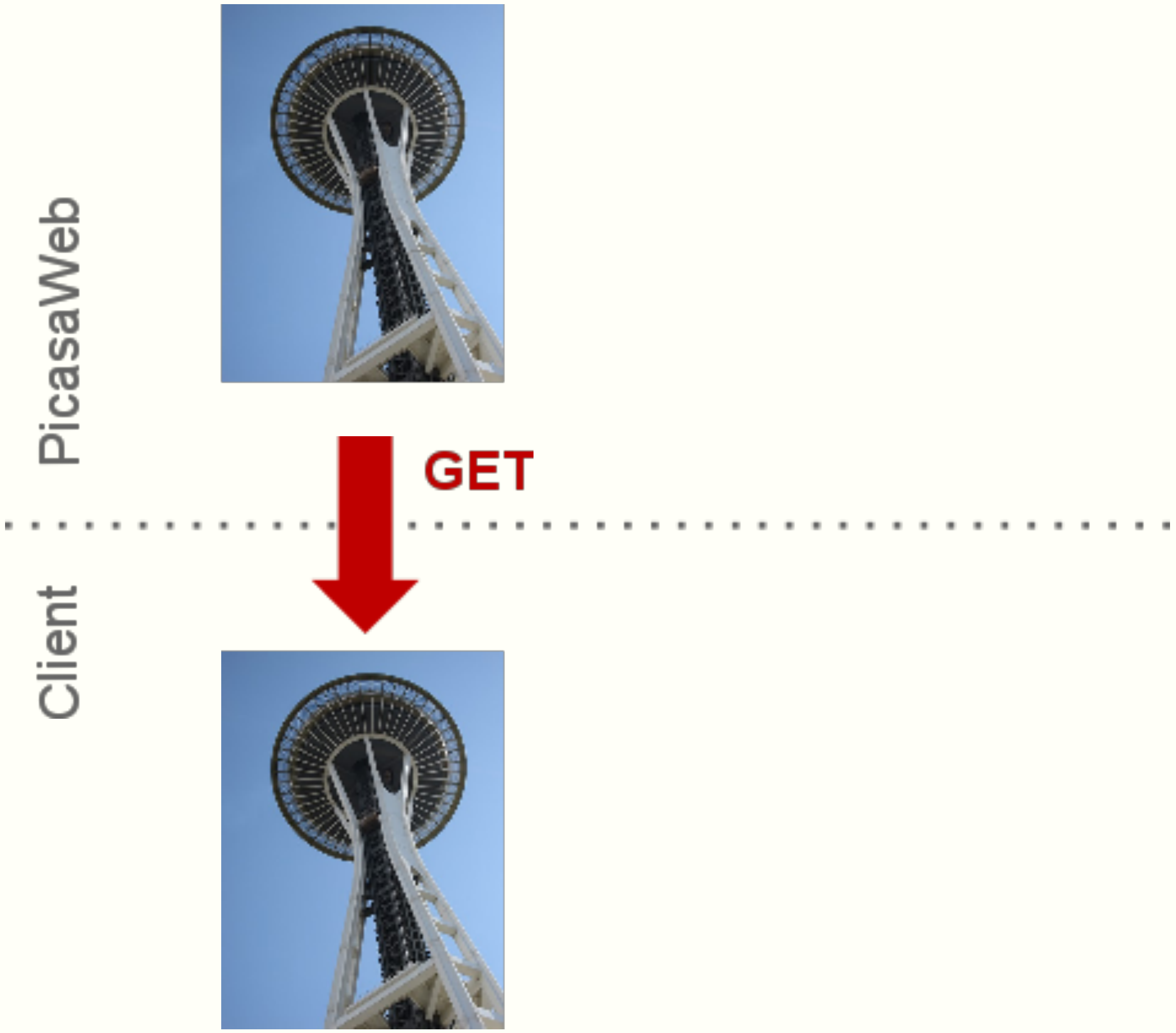
Calling Styles

- **Issue: REST can be Awkward**

PicasaWeb
Client

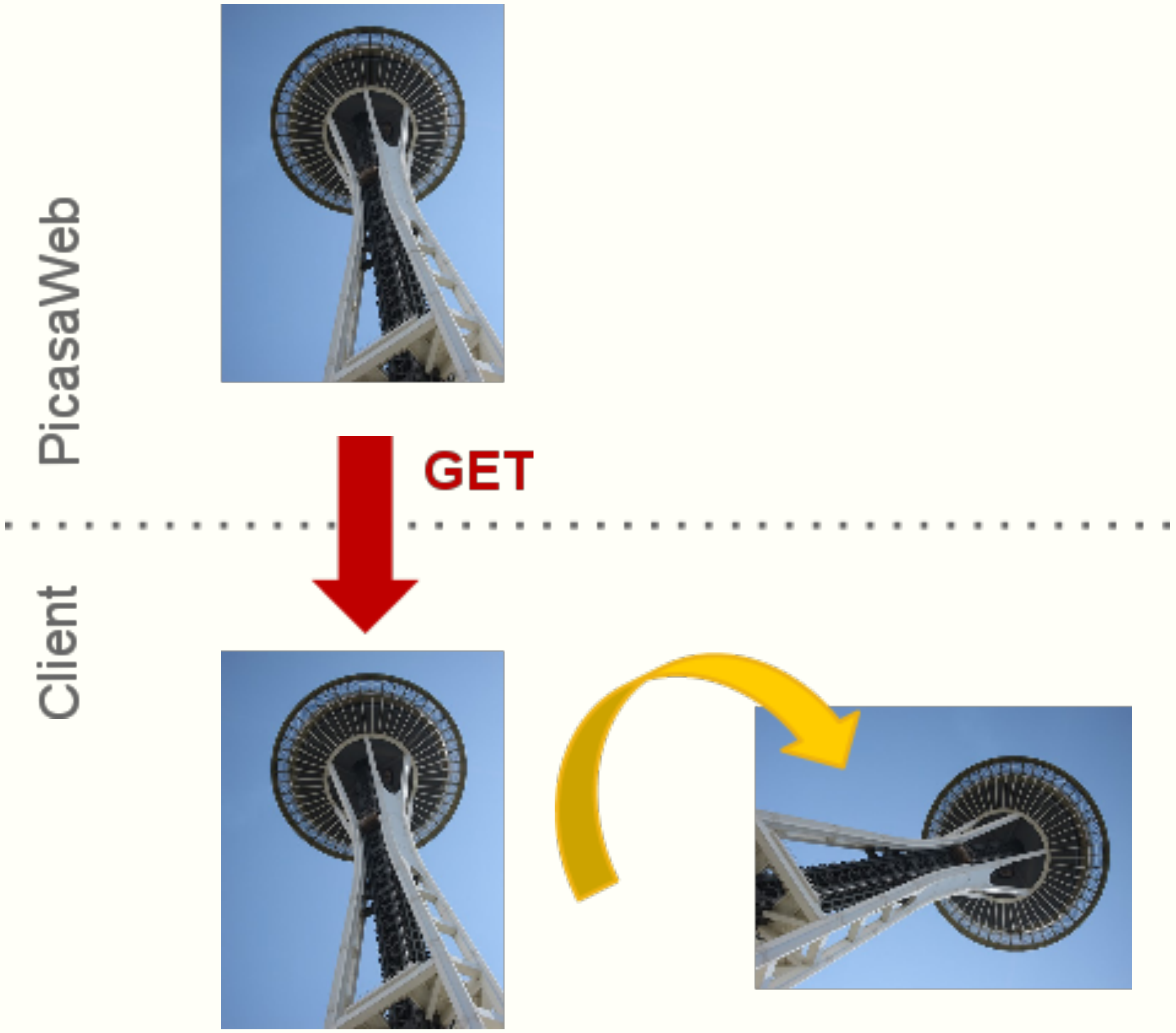
Calling Styles

- **Issue: REST can be Awkward**



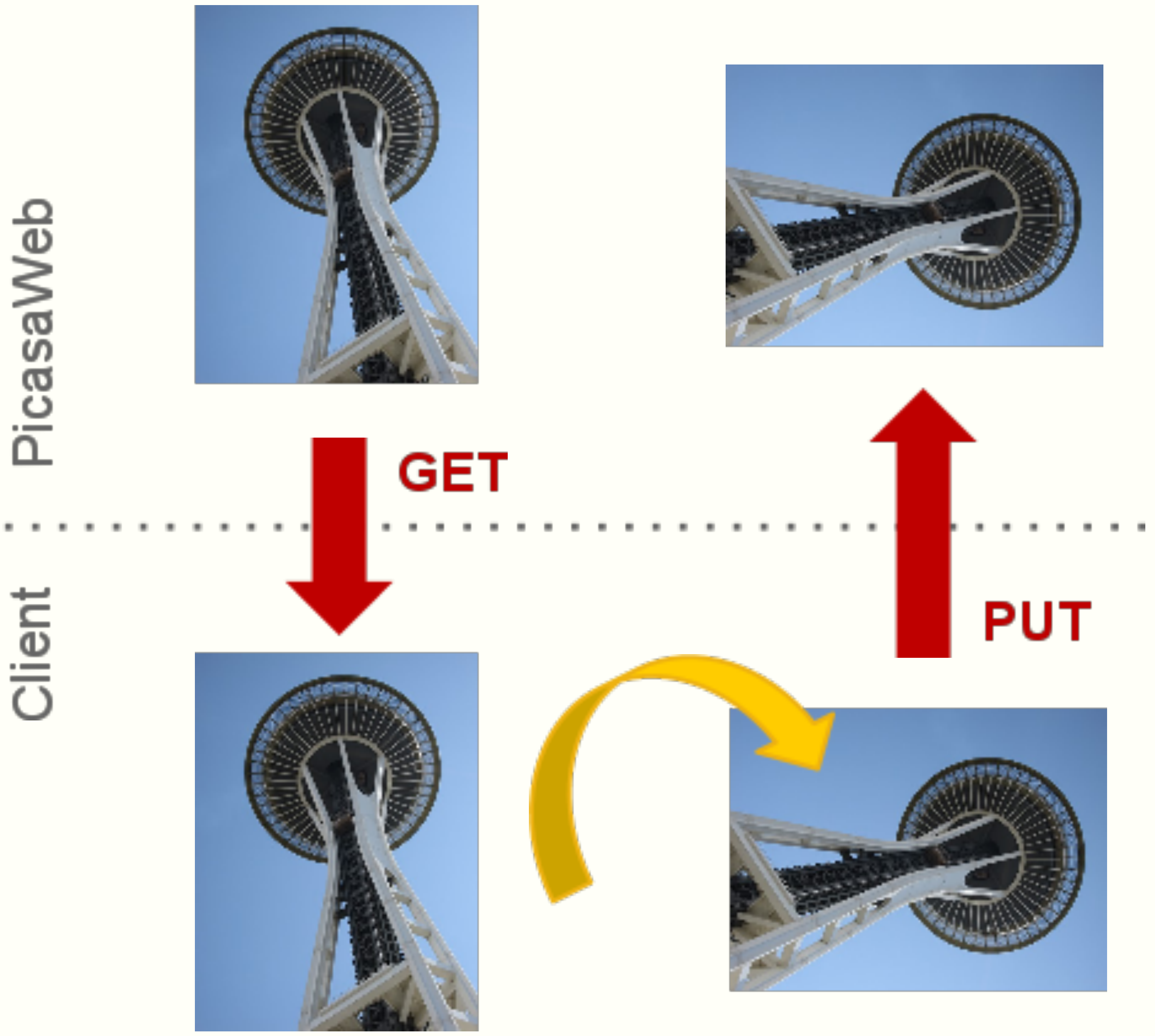
Calling Styles

- **Issue: REST can be Awkward**



Calling Styles

- **Issue: REST can be Awkward**



Calling Styles

- **Solution: Augment REST with Custom Verbs**

Calling Styles

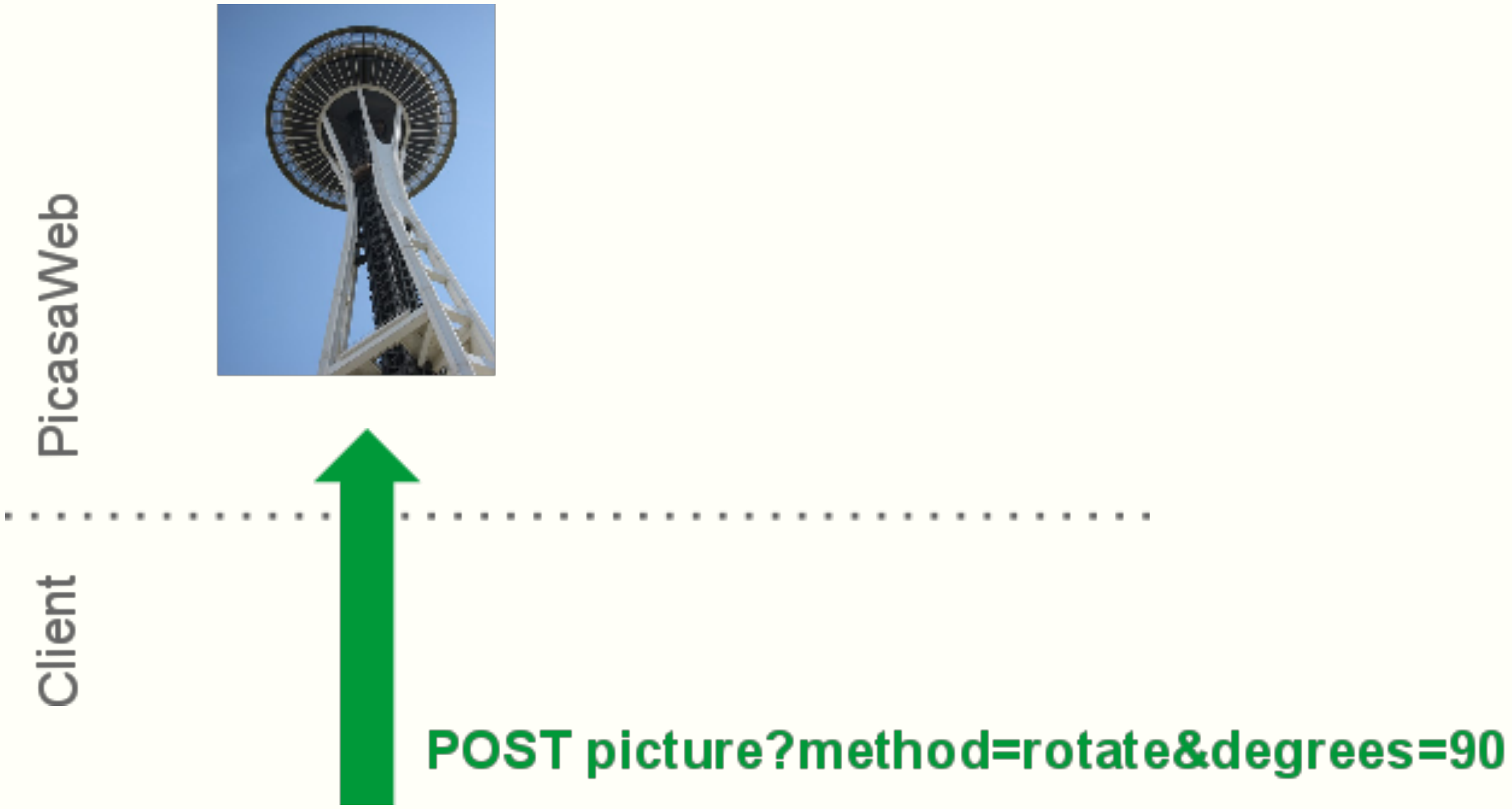
- **Solution: Augment REST with Custom Verbs**

PicasaWeb

Client

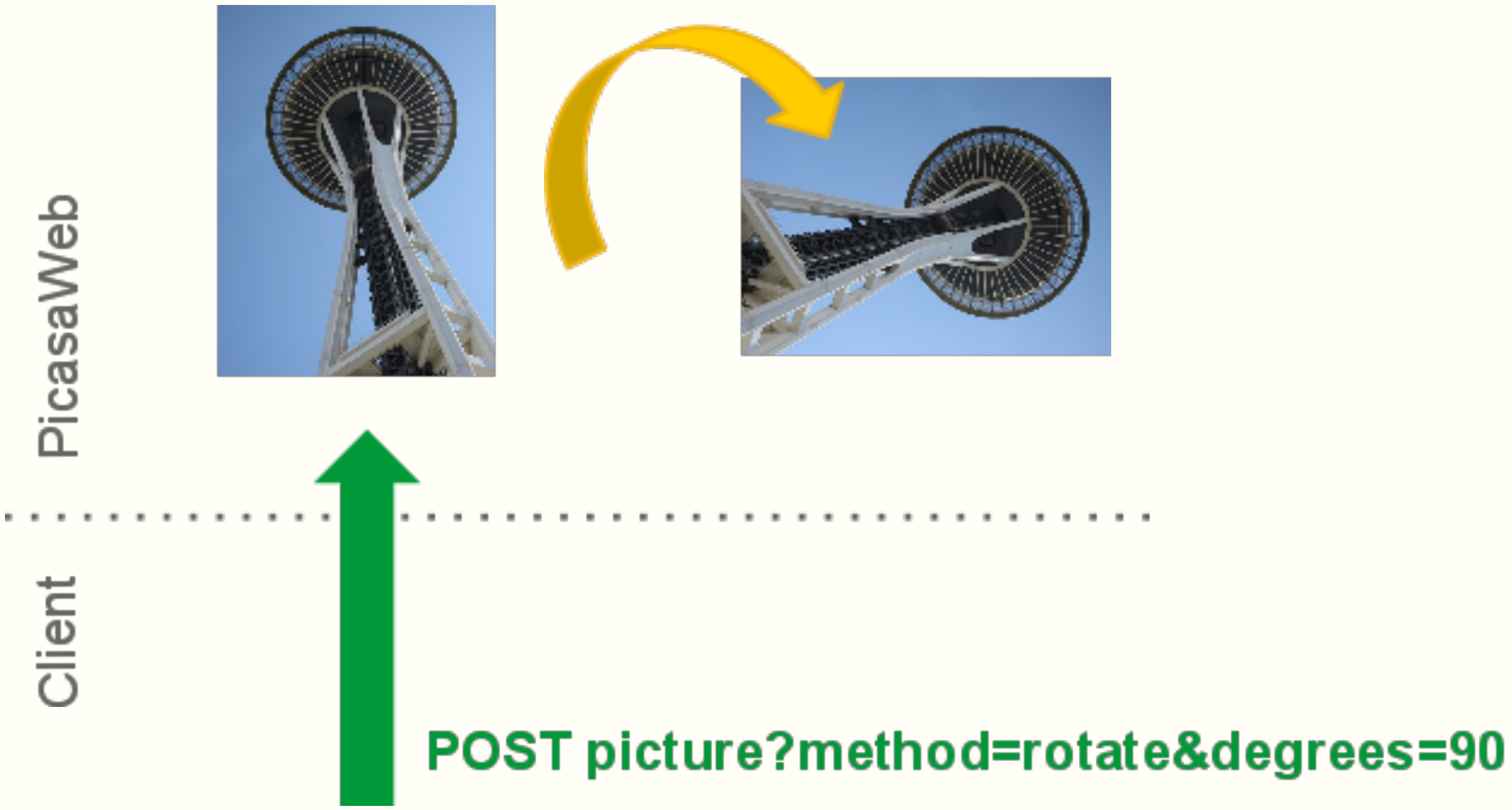
Calling Styles

- **Solution: Augment REST with Custom Verbs**



Calling Styles

- **Solution: Augment REST with Custom Verbs**



POST picture?method=rotate&degrees=90

Calling Styles

- **Solution: Augment REST with Custom Verbs**
- Example: Two ways to mark a task as "done"

Calling Styles

- **Solution: Augment REST with Custom Verbs**
- Example: Two ways to mark a task as "done"
- **The RESTful way**
  - GET /tasks/@me/{taskId}
  - modify resource on client, set the "done" bit
  - PUT /tasks/@me/{taskId}

Calling Styles

- **Solution: Augment REST with Custom Verbs**
- Example: Two ways to mark a task as "done"
- The RESTful way
  - GET /tasks/@me/{taskId}
  - modify resource on client, set the "done" bit
  - PUT /tasks/@me/{taskId}
- **Using custom verbs**
  - POST /tasks/@me/{taskId}?method=markDone

Calling Styles

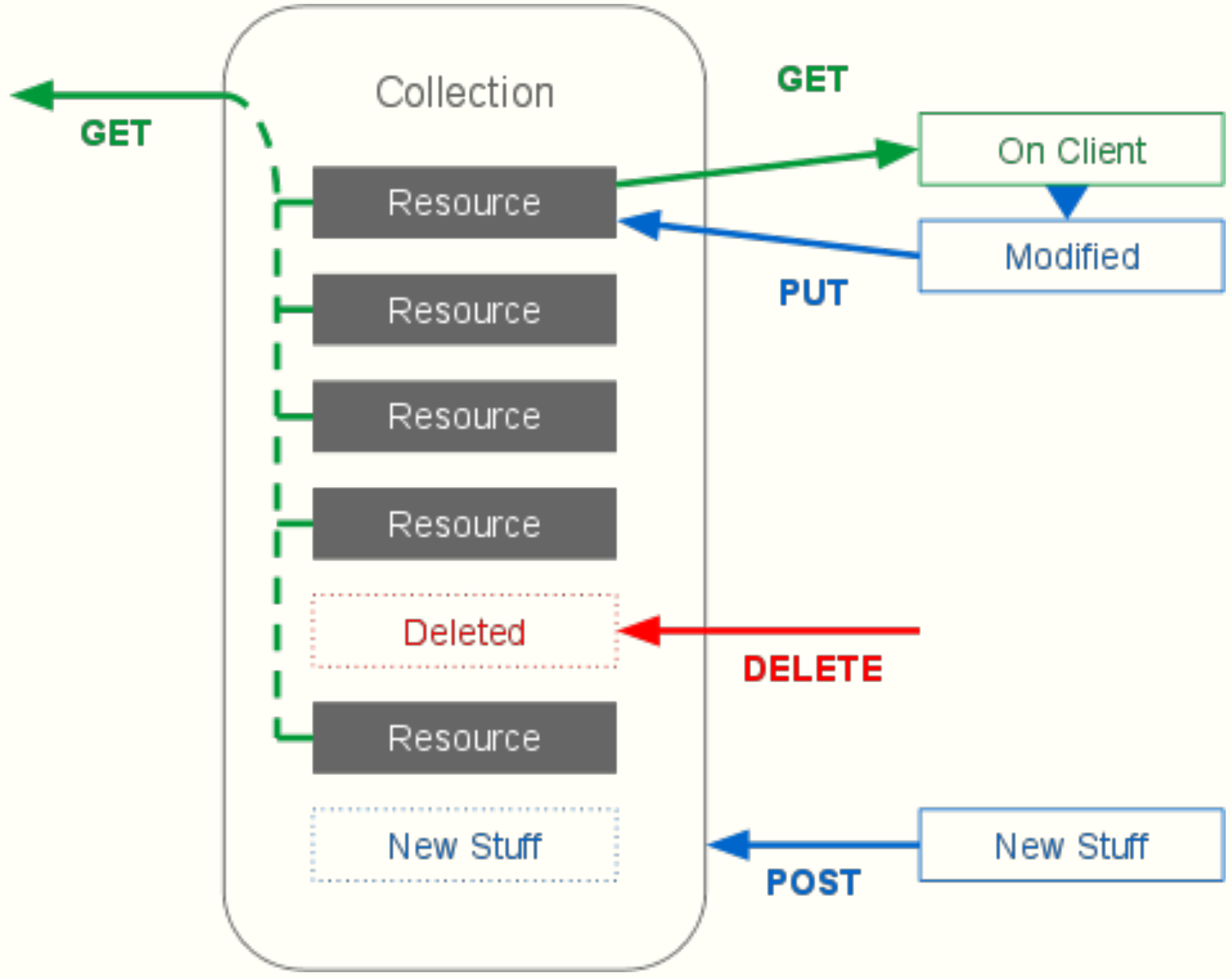
- **Issue: RPC is the basis of many API Standards**

Calling Styles

- **Solution: Parallel REST and JSON-RPC interface**

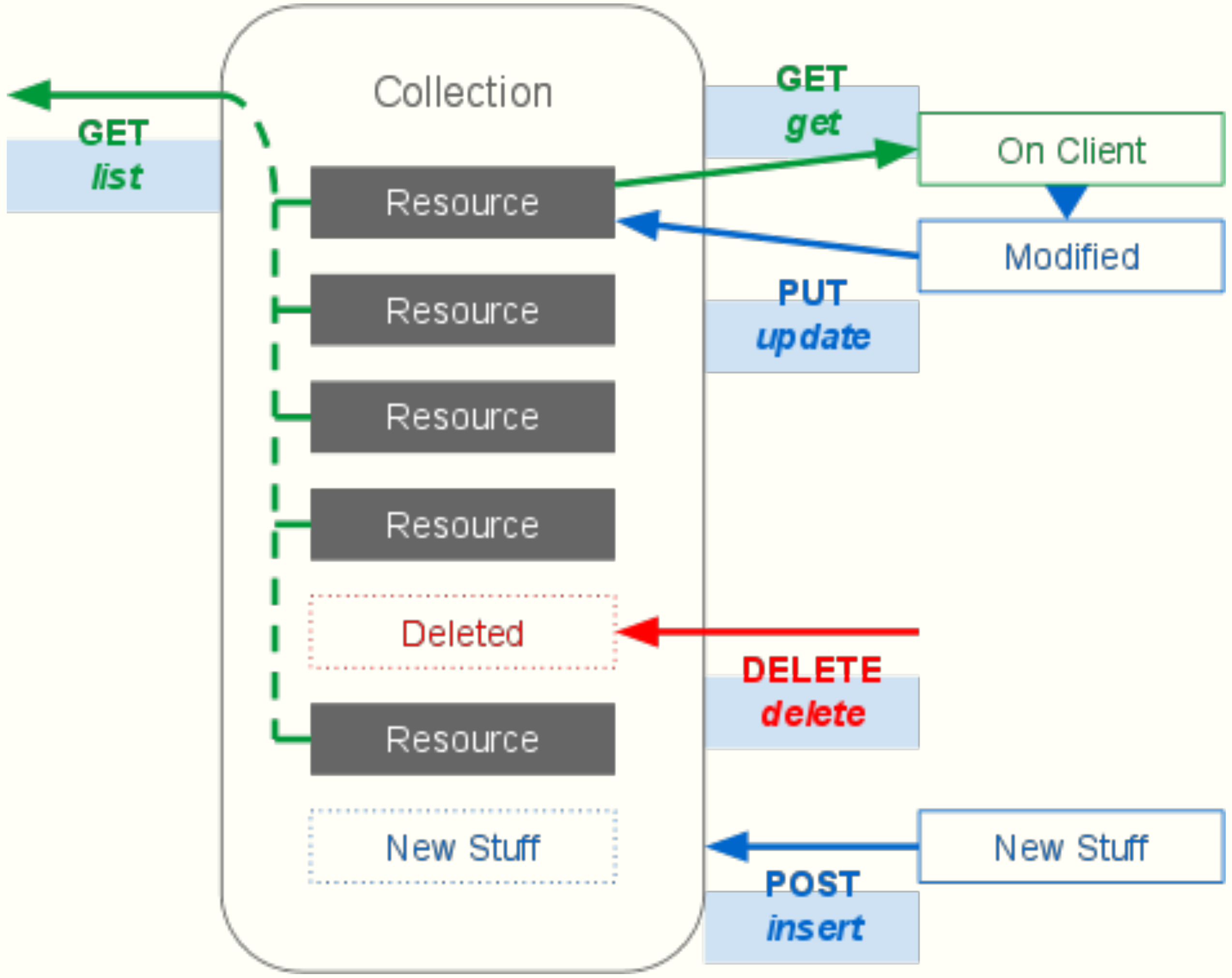
Calling Styles

- **Solution: Parallel REST and JSON-RPC interface**



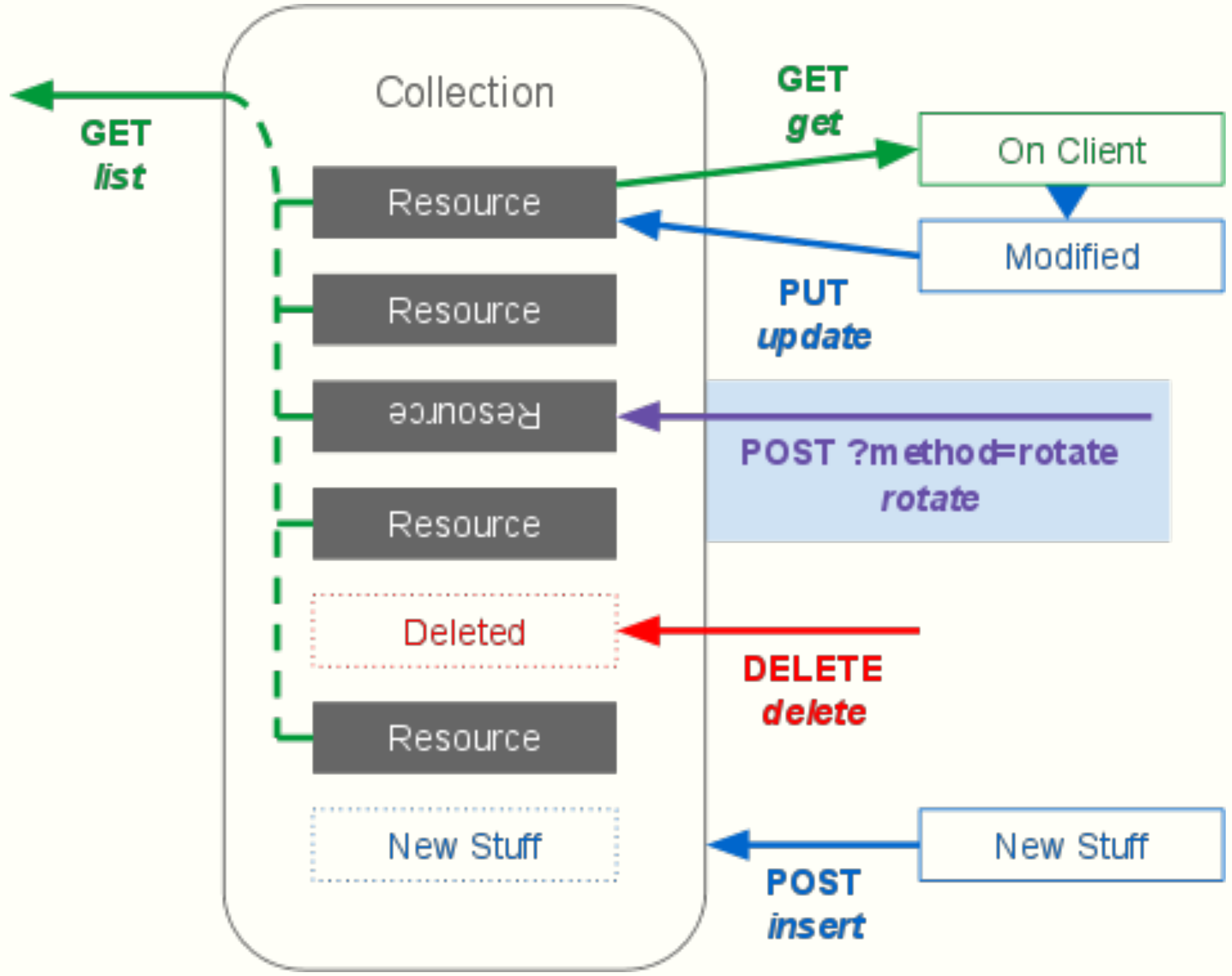
Calling Styles

- **Solution: Parallel REST and JSON-RPC interface**
- Parallel calling styles for common methods



Calling Styles

- **Solution: Parallel REST and JSON-RPC interface**
- Parallel calling styles for common methods, and custom verbs



Client Libraries

Client Libraries

- **Issue: Client libraries don't stay on the cutting edge**

Client Libraries

- **Solution: Dynamic Discovery**

Client Libraries

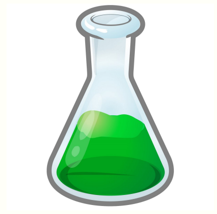
- **Solution: Dynamic Discovery**
- **Discovery Document**
    - JSON object
    - Describes resources, URLs, verbs, parameters, (schemas)
    - Always up to date

Client Libraries

- **Solution: Dynamic Discovery**
- Discovery Document
    - JSON object
    - Describes resources, URLs, verbs, parameters, (schemas)
    - Always up to date
- **Discovery is just another API**

http://www.googleapis.com/directory/v0.1/describe?api=discovery&apiVersion=v1

bit.ly/buzzdiscovery



Client Libraries

- **Solution: "Generic" Client Libraries**

Client Libraries

- **Solution: "Generic" Client Libraries**
- **Discover resources, URL templates, verbs**
  - No more scraping URLs from doc

Client Libraries

- **Solution: "Generic" Client Libraries**
- Discover resources, URL templates, verbs
    - No more scraping URLs from doc
- **Use simple classes to represent resources**
    - Create POJOs to map JSON

Client Libraries

- **Solution: "Generic" Client Libraries**
- Discover resources, URL templates, verbs
  - No more scraping URLs from doc
- Use simple classes to represent resources
  - Create POJOs to map JSON
- **Make advanced features easy(er)**
  - Batching, Async, Partial Get and Update

Client Libraries

- **Solution: "Generic" Client Libraries**
- Discover resources, URL templates, verbs
  - No more scraping URLs from doc
- Use simple classes to represent resources
  - Create POJOs to map JSON
- Make advanced features easy(er)
  - Batching, Async, Partial Get and Update
- **Works on multiple platforms**
  - Java client works on servers, AppEngine, Android
  - JavaScript client works on web pages, gadgets, AppScript

Client Libraries

- **Solution: "Generic" Client Libraries**
- Discover resources, URL templates, verbs
  - No more scraping URLs from doc
- Use simple classes to represent resources
  - Create POJOs to map JSON
- Make advanced features easy(er)
  - Batching, Async, Partial Get and Update
- Works on multiple platforms
  - Java client works on servers, AppEngine, Android
  - JavaScript client works on web pages, gadgets, AppScript
- **Release once ... Works with Any API**

Java Client Demonstration

Client Libraries

- **Solution: "Generic" Client Libraries**

Client Libraries

- **Solution: "Generic" Client Libraries**
- **Why we like this client sample**
  - Resource URLs are discovered dynamically
  - Very little Buzz specific code
  - Runs on Android

Client Libraries

- **Solution: "Generic" Client Libraries**
- Why we like this client sample
  - **Resource URLs are discovered dynamically**
  - **Very little Buzz specific code**
  - **Runs on Android**
- **Go check it out yourself!**
- Sample: bit.ly/iobuzz
- Library: bit.ly/javalib

The Future Google APIs

The Future Google APIs

- **Issues we've noticed in 5 years of building APIs**

- **How we're fixing those issues**

The Future Google APIs

- **Issues we've noticed in 5 years of building APIs**
    - Resources can be verbose
- **How we're fixing those issues**
    - Allow operations on partial data

The Future Google APIs

- **Issues we've noticed in 5 years of building APIs**
    - Resources can be verbose
    - XML isn't easy on all platforms

- **How we're fixing those issues**
    - Allow operations on partial data
    - Provide multiple read-write formats

The Future Google APIs

- **Issues we've noticed in 5 years of building APIs**
  - Resources can be verbose
  - XML isn't easy on all platforms
  - REST can be awkward

- **How we're fixing those issues**
  - Allow operations on partial data
  - Provide multiple read-write formats
  - Augment REST with custom verbs, provide REST and RPC

The Future Google APIs

- **Issues we've noticed in 5 years of building APIs**
  - Resources can be verbose
  - XML isn't easy on all platforms
  - REST can be awkward
  - Client libraries don't stay on the cutting edge

- **How we're fixing those issues**
  - Allow operations on partial data
  - Provide multiple read-write formats
  - Augment REST with custom verbs, provide REST and RPC
  - Dynamic discovery and "generic" client libraries

How Google Builds APIs

- Google APIs 101
- Making Future APIs Awesome
- **How Google *Really* Builds APIs**
- Questions and Comments

How Google *Really* Builds APIs

How Google *Really* Builds APIs

- **Implement Internal Service**
  - Define abstract resource (using protocol buffers)
  - Define collections and verbs (using protocol buffer RPC)

How Google *Really* Builds APIs

- Implement Internal Service
  - Define abstract resource (using protocol buffers)
  - Define collections and verbs (using protocol buffer RPC)
- **Configure the API Stack**
  - Map REST paths, RPC methods, query parameters
  - Add common functionality: auth, caching, logging, ...

How Google *Really* Builds APIs

- Implement Internal Service
  - Define abstract resource (using protocol buffers)
  - Define collections and verbs (using protocol buffer RPC)
- Configure the API Stack
  - Map REST paths, RPC methods, query parameters
  - Add common functionality: auth, caching, logging, ...
- **Write Output Templates**
  - Set up external data representation
  - JSON, Atom, XML, ...

API Configuration Tool Demonstration

How Google *Really* Builds APIs

- You just saw the simple three steps:
    - Service implemented
    - API stack configured
    - Output templates written

How Google Builds APIs

- Google APIs 101
- Making Future APIs Awesome
- How Google *Really* Builds APIs
- **Questions and Comments**

http://bit.ly/apiwave