

Designing Rich API Clients at Scale

Jeff Ching

Code BEAM SF
March 16, 2018





Google Cloud Platform

Languages, Optimizations, and Libraries

Languages, Optimizations, and Libraries

Designing Rich API Clients at Scale

- What's the problem?
- Case studies
- What can you do?

What we won't cover

- Scaling your application
- Designing APIs

Networked APIs

Application Programming Interfaces that operate across a network of computers. They communicate using network protocols including HTTP, and are frequently produced by different organizations than the ones that consume them.

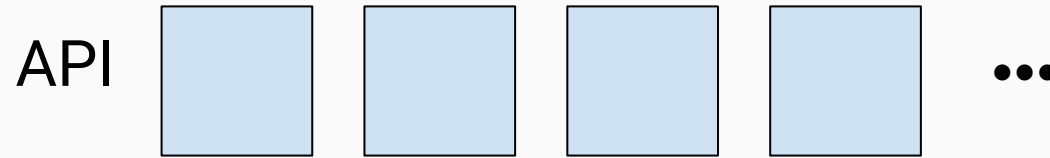
Google API Design Guide: Glossary
<https://cloud.google.com/apis/design/glossary>

What's the problem?

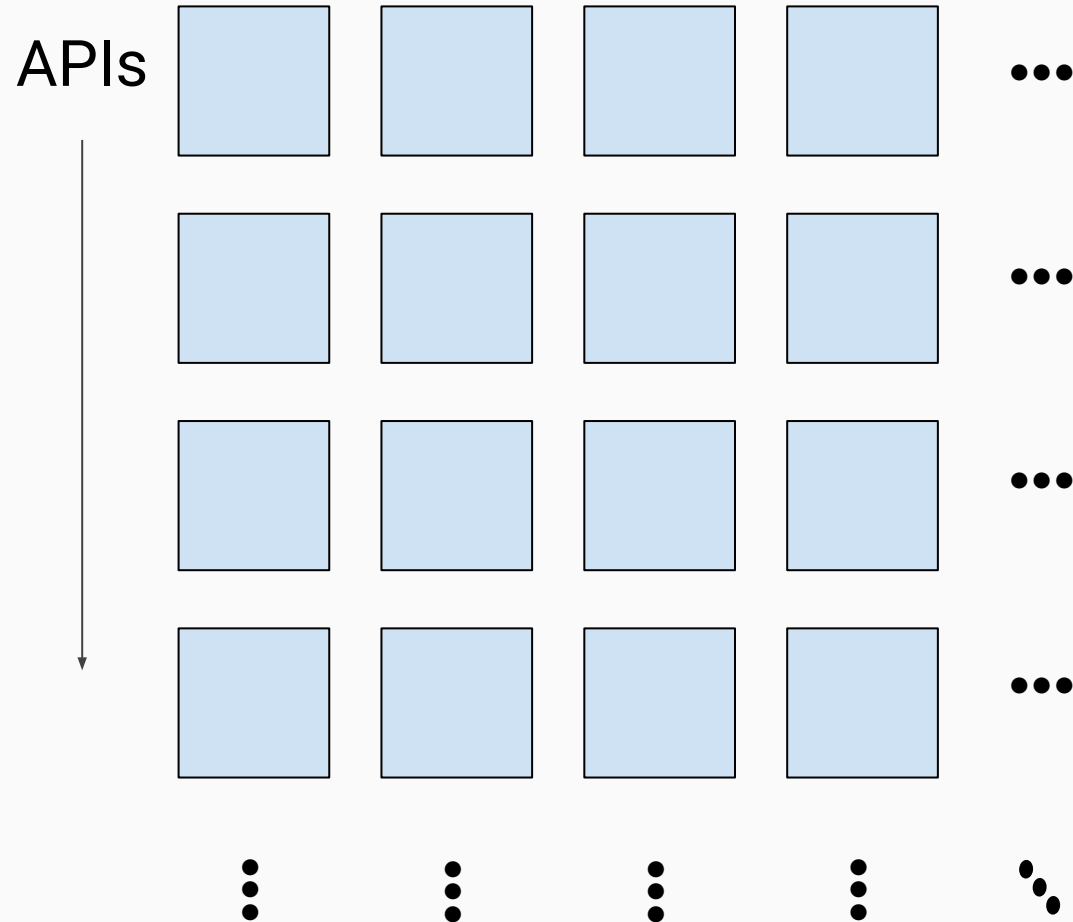
API



Languages 



Languages 

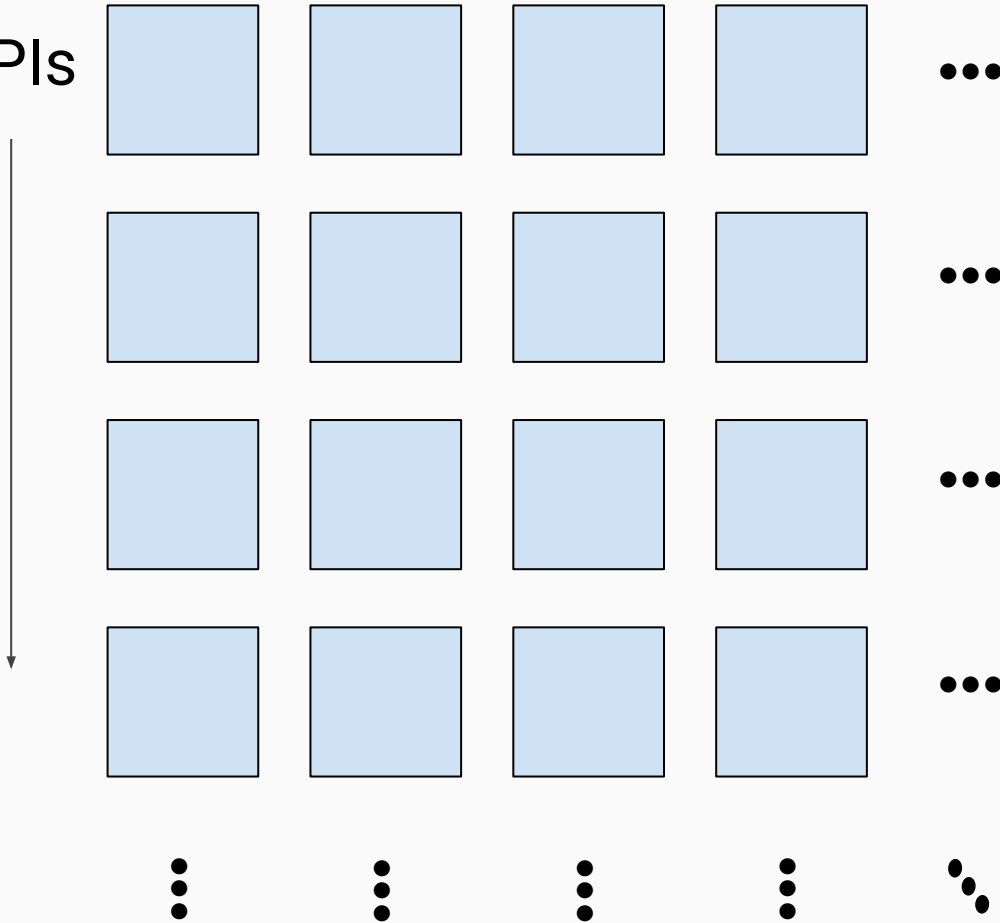


Y

Languages →

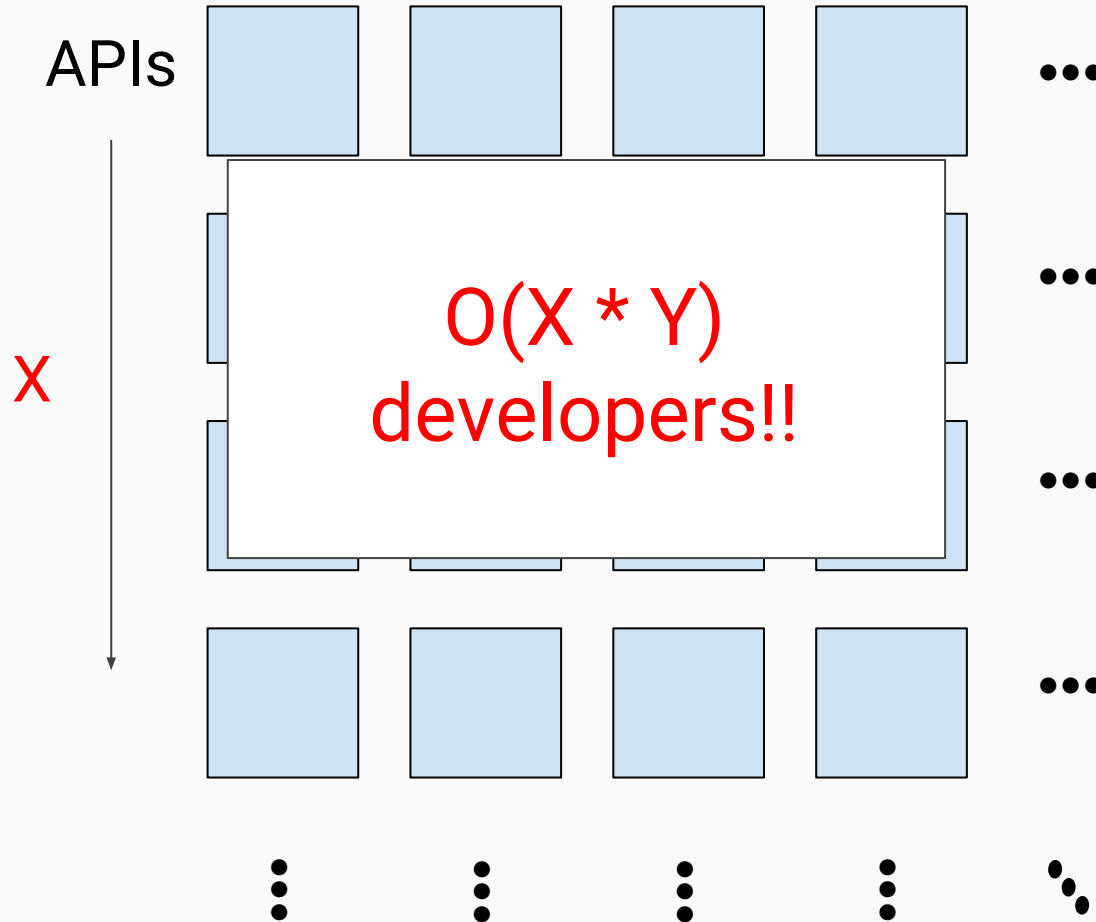
APIs

X



Y

Languages \longrightarrow



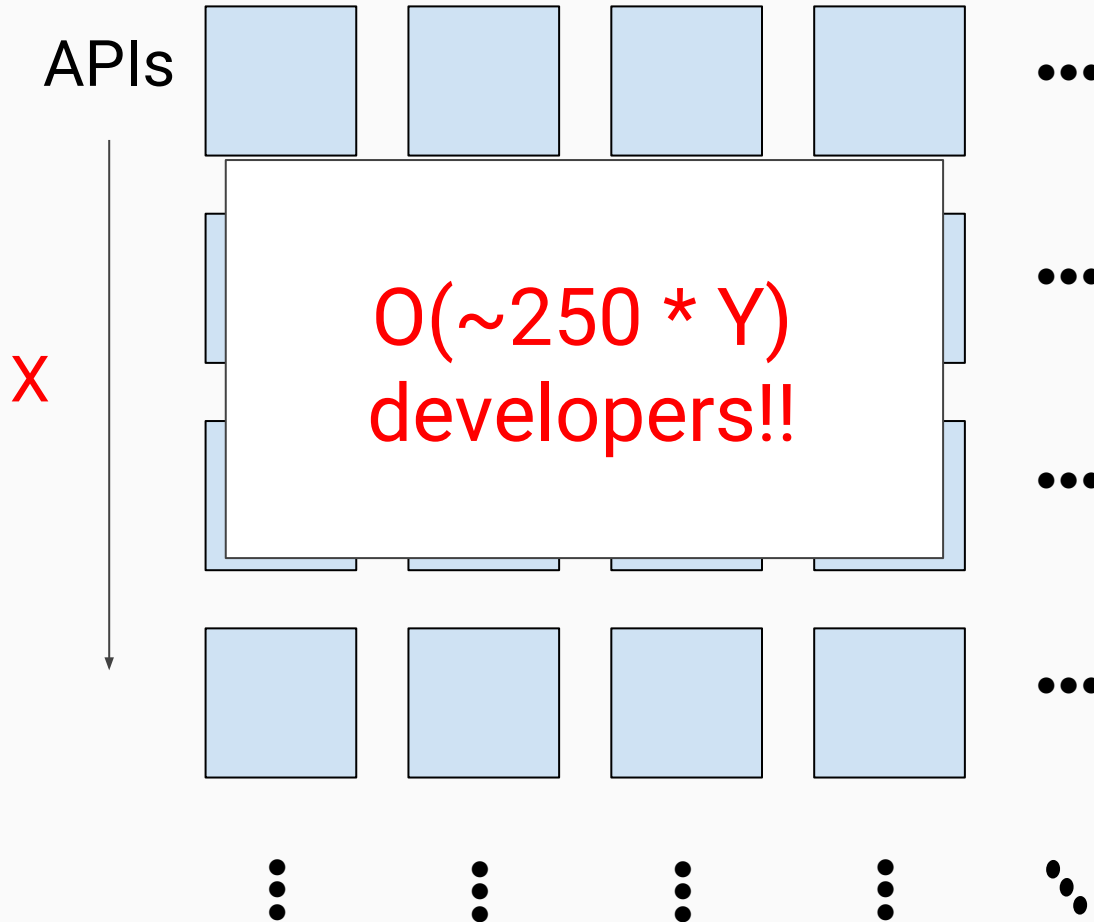
X

APIs

$O(X * Y)$
developers!!

Y

Languages \longrightarrow



Code Generation (Codegen)

Code Generation (Codegen)

- Scalability

Code Generation (Codegen)

- Scalability
- Leverage

Code Generation (Codegen)

- Scalability
- Leverage
- Consistency

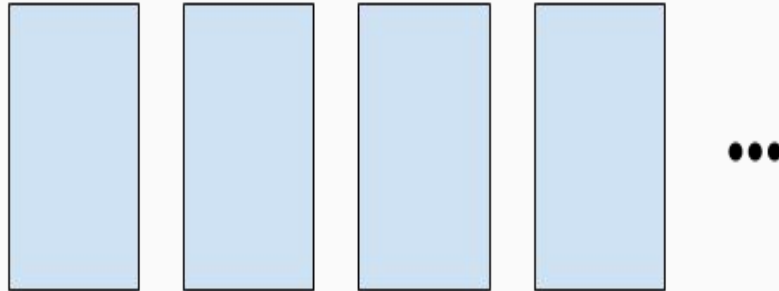
Code Generation (Codegen)

- Scalability
- Leverage
- Consistency
- Safety

Y

Languages \longrightarrow

Template



Y

Languages 

Templat



...

What's in an API Client?

Components

API Description		
<ul style="list-style-type: none">● Data Types● Endpoints		

API Description	Transport/Protocol	
<ul style="list-style-type: none">● Data Types● Endpoints	<ul style="list-style-type: none">● HTTP/REST/JSON● Binary Protocol (gRPC, Thrift, etc)	

API Description	Transport/Protocol	Authentication
<ul style="list-style-type: none">● Data Types● Endpoints	<ul style="list-style-type: none">● HTTP/REST/JSON● Binary Protocol (gRPC, Thrift, etc)	<ul style="list-style-type: none">● HTTP Basic● OAuth● Public/Private Key

Case Study 1: Google APIs - Dynamic Clients

Google Discovery

```
{
  "kind": "discovery#restDescription",
  "discoveryVersion": "v1",
  "id": string,
  "name": string,
  "version": string,
  "revision": string,
  "title": string,
  "description": string,
  "icons": {
    "x16": string,
    "x32": string
  },
  "documentationLink": string,
  "labels": [
    string
  ],
  "protocol": "rest",
  "baseUrl": string,
  "basePath": string,
  "rootUrl": string,
  "servicePath": string,
  "batchPath": "batch",
  "parameters": {
```

Goals

- Scalability

Goals

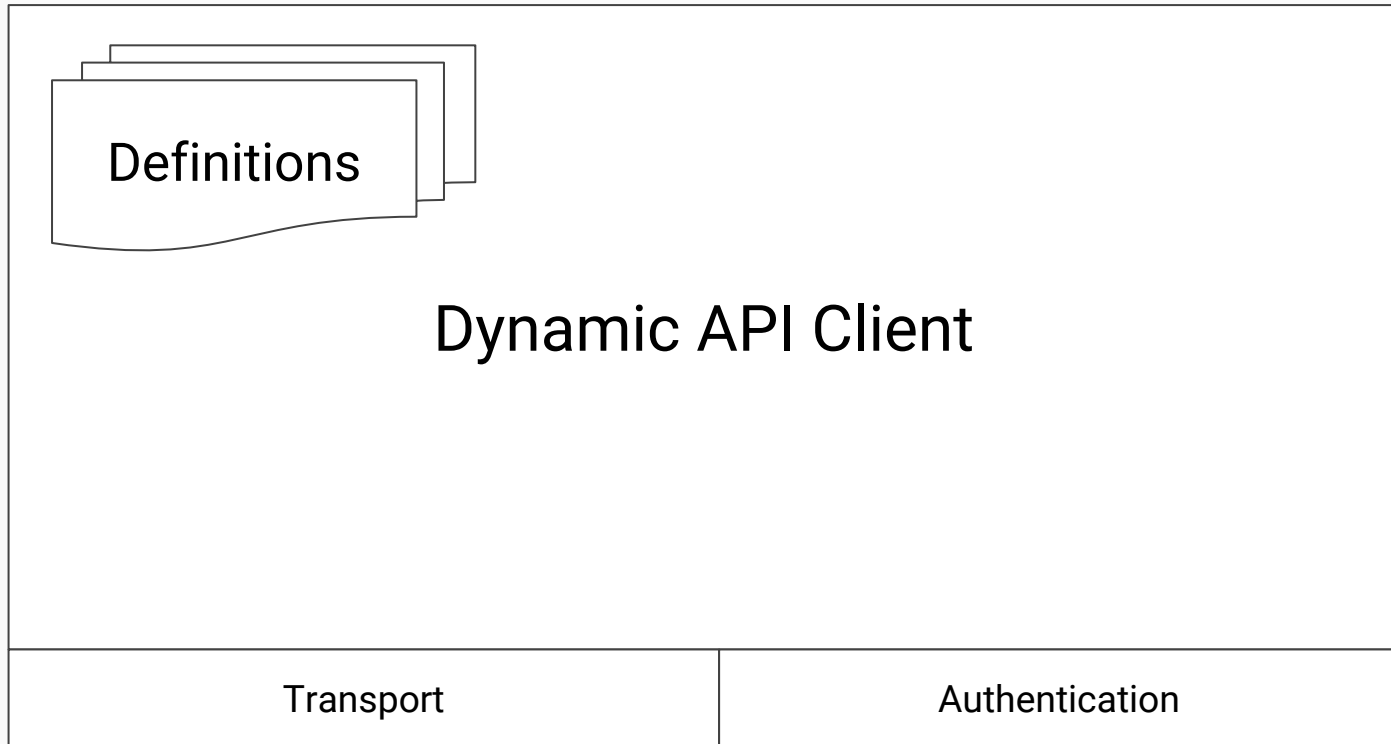
- Scalability
- Automation

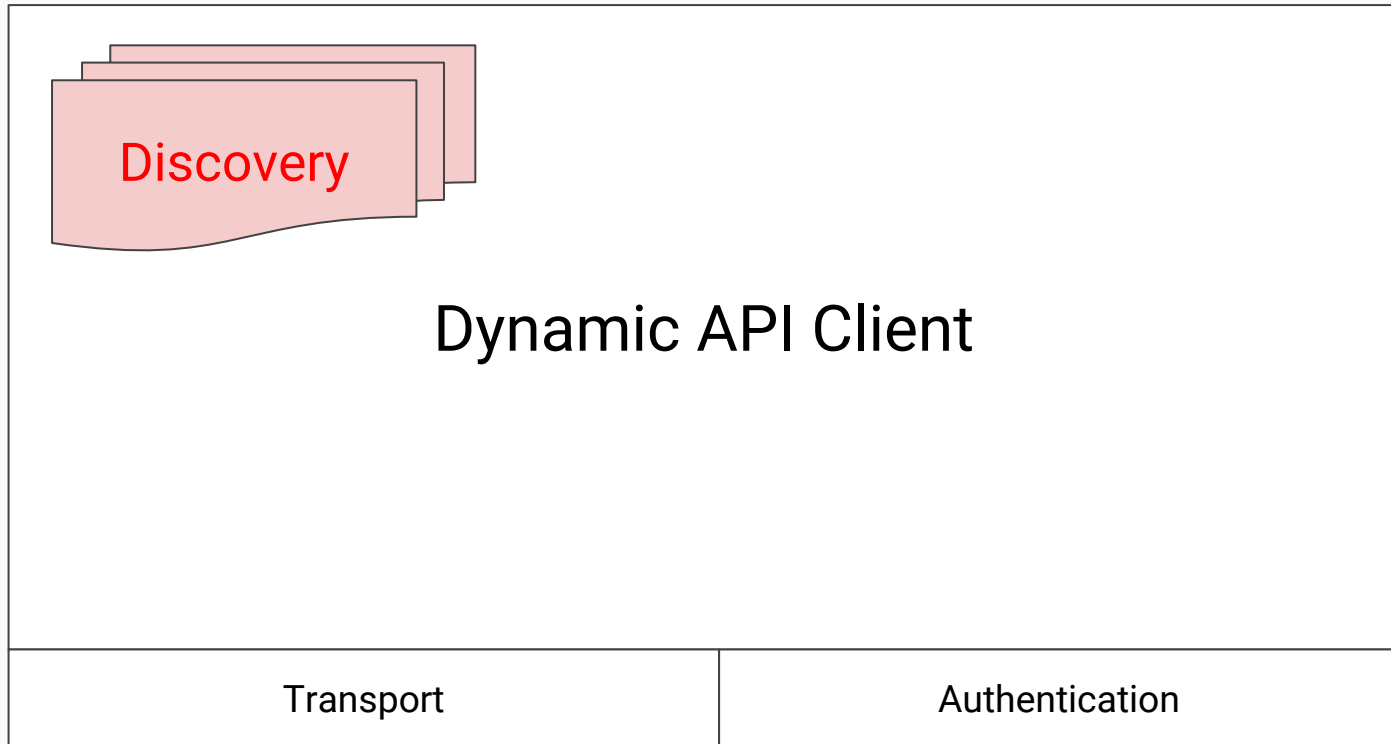
Goals

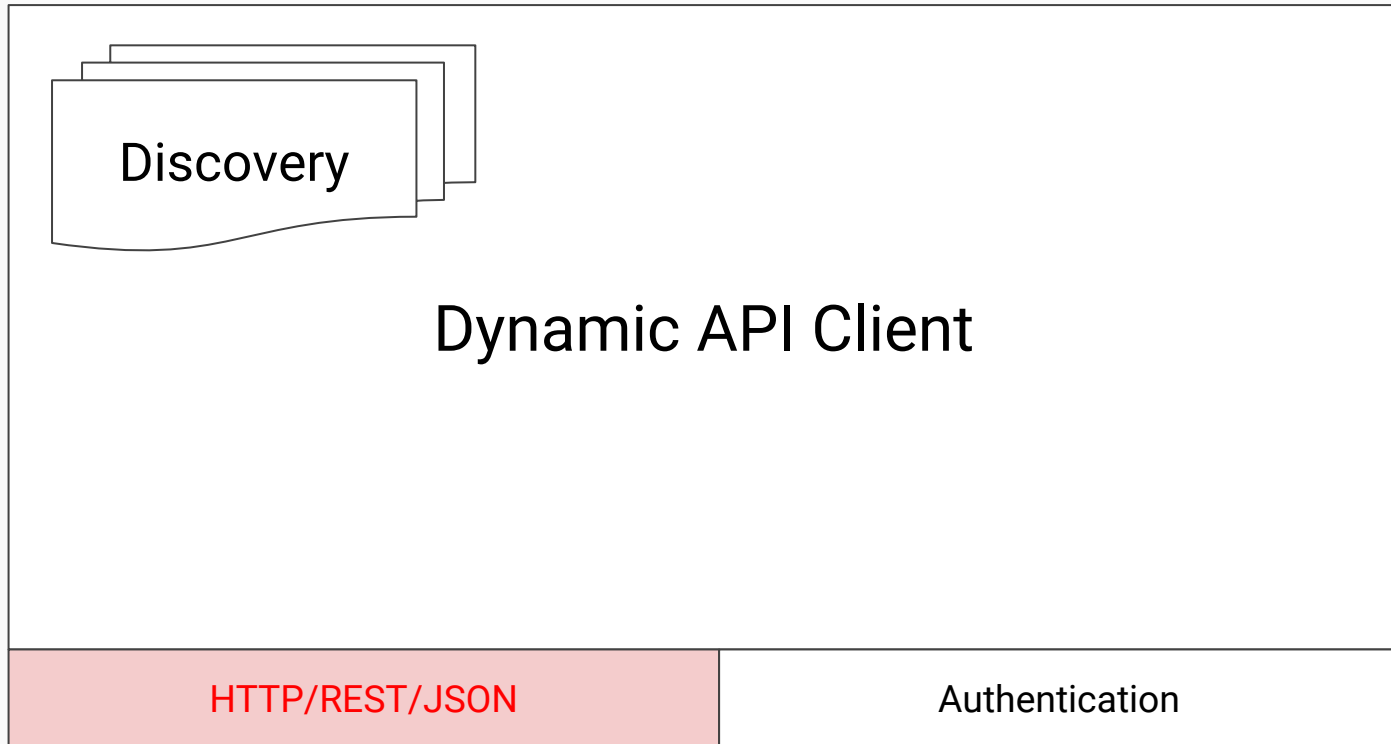
- Scalability
- Automation
- Transparency

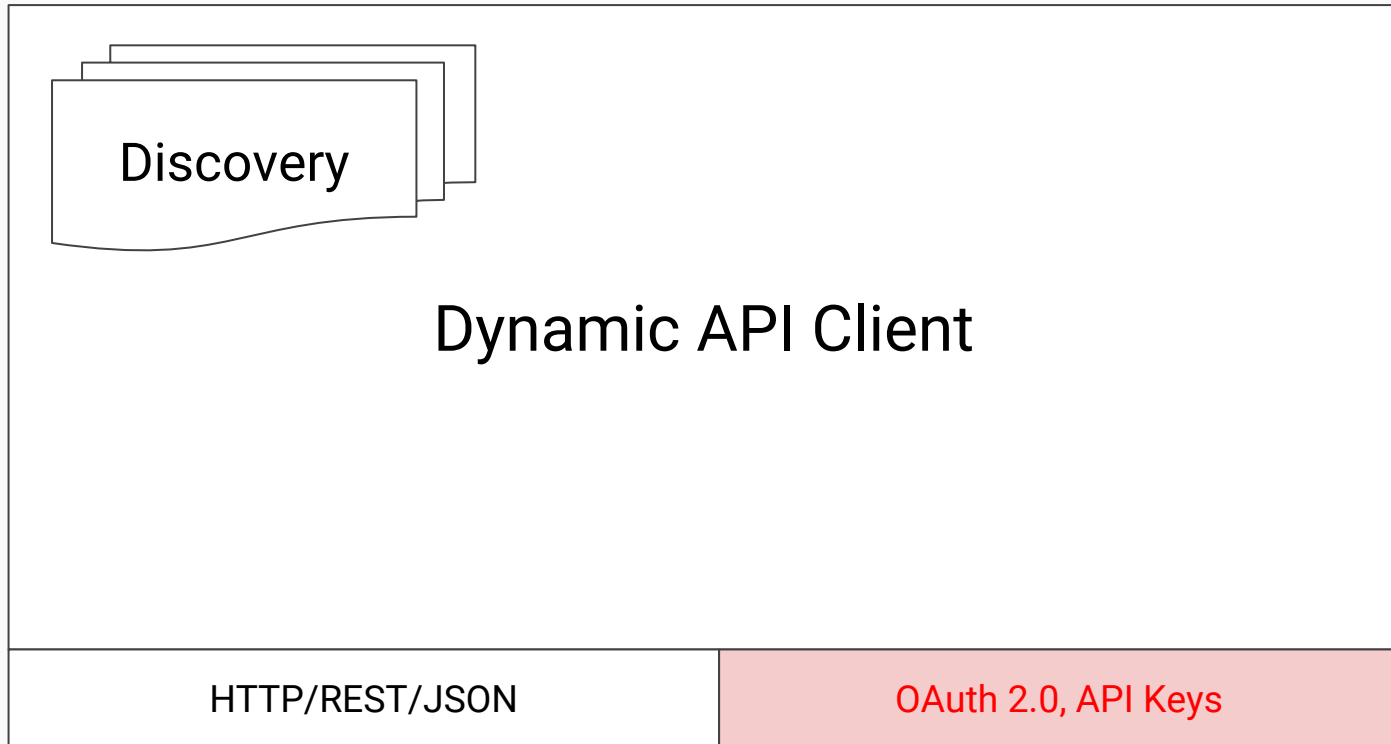
Goals

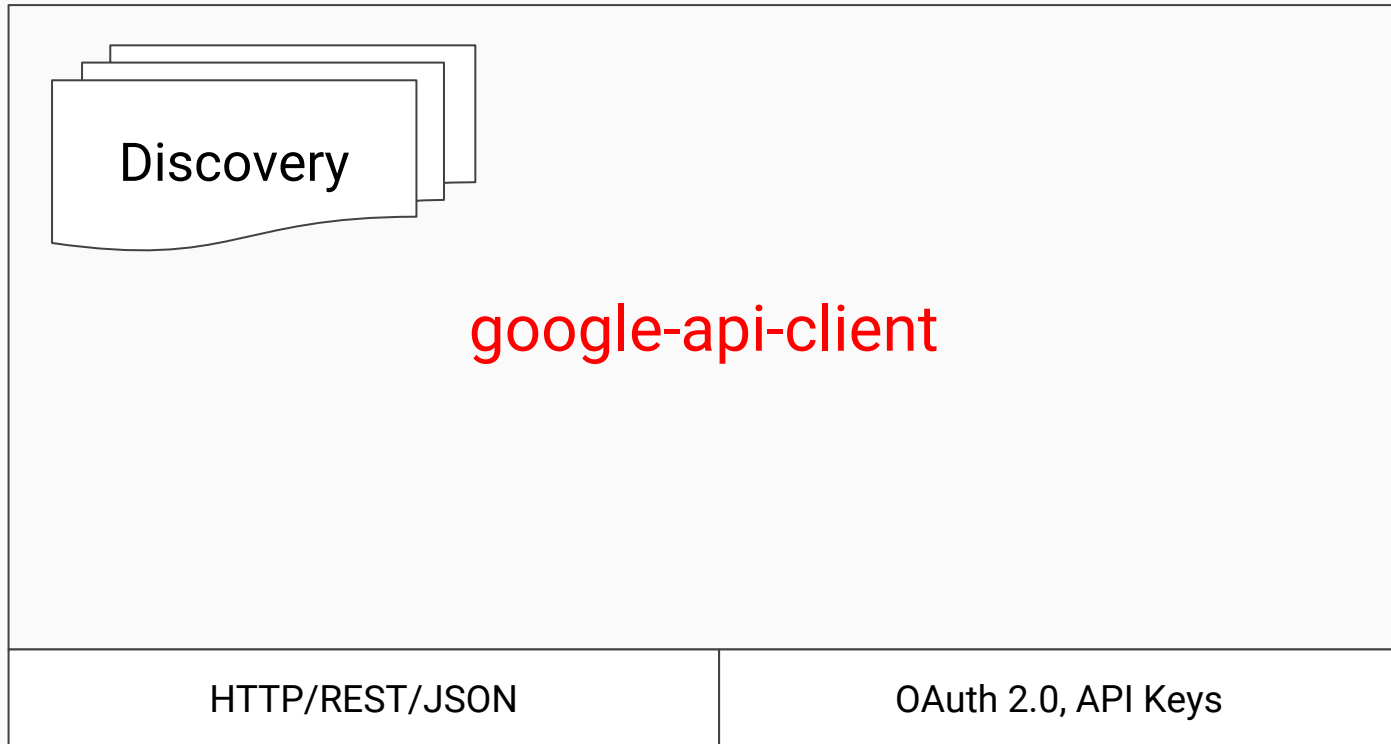
- **Scalability**
- Automation
- Transparency











```
1 # build client for service
2 specification = File.read "/path/to/spec.json"
3 builder = GoogleApis.builder_for specification
4 service = builder.service_for "storage"
5
6 # insert object
7 obj = %{field: "value"}
8 service.insert_object obj
```

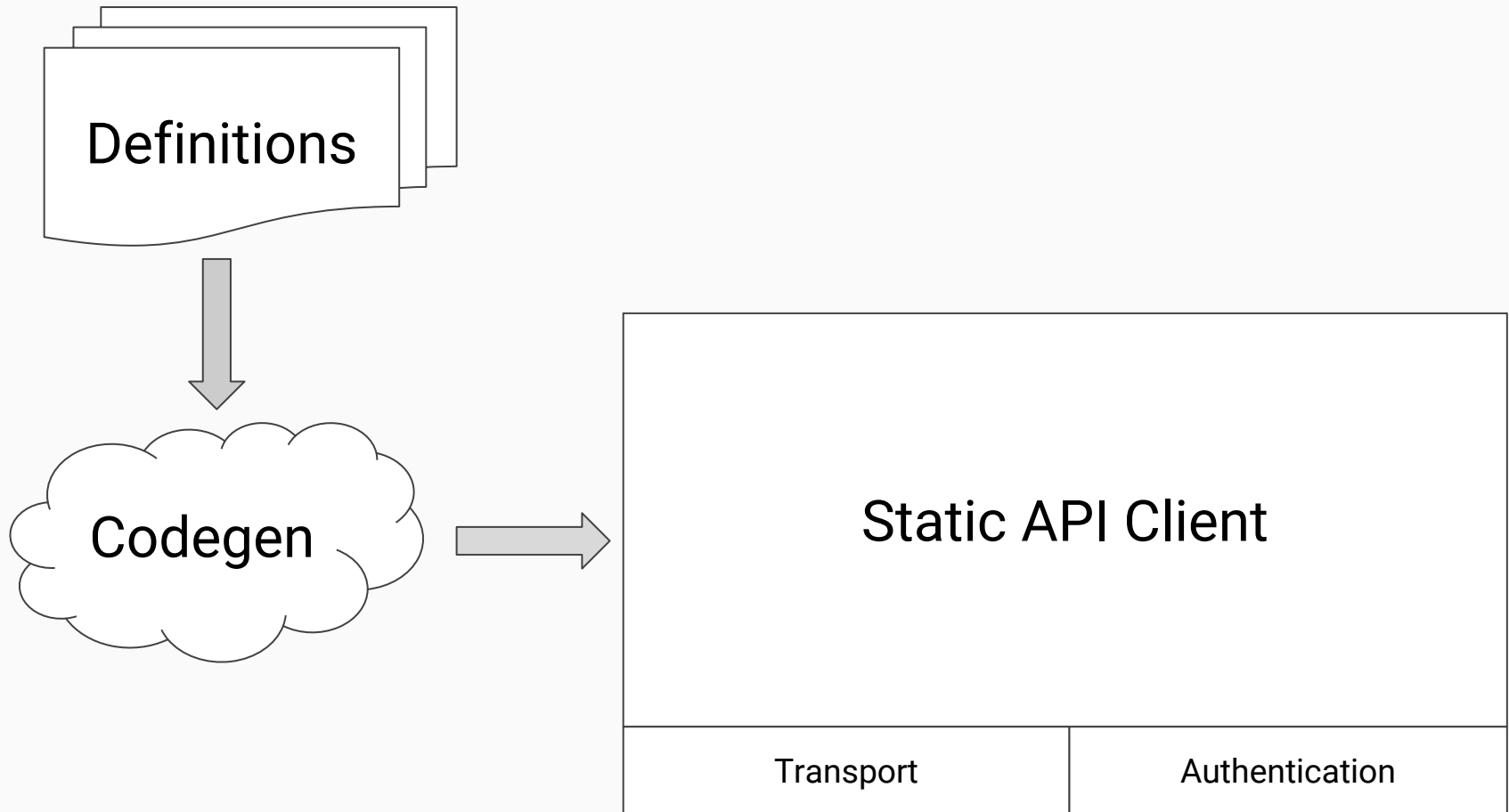
```
1 # build client for service
   specification = file.read("/to/spec.json")
3 bus = ...
5
6 # insert
7 obj = %{"field": "value"}
8 service.insert_object obj
```

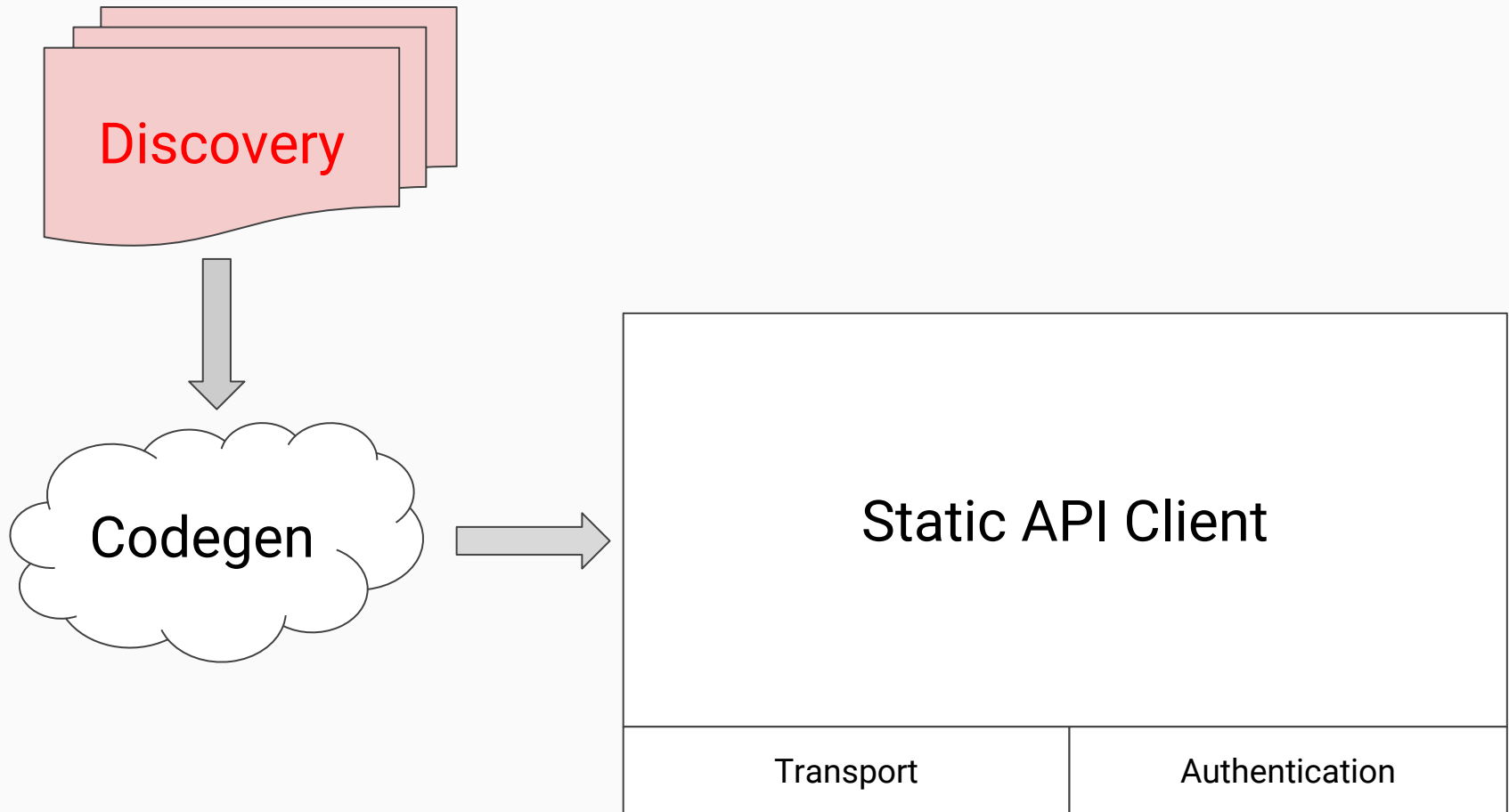
Pro Tip: Avoid metaprogramming

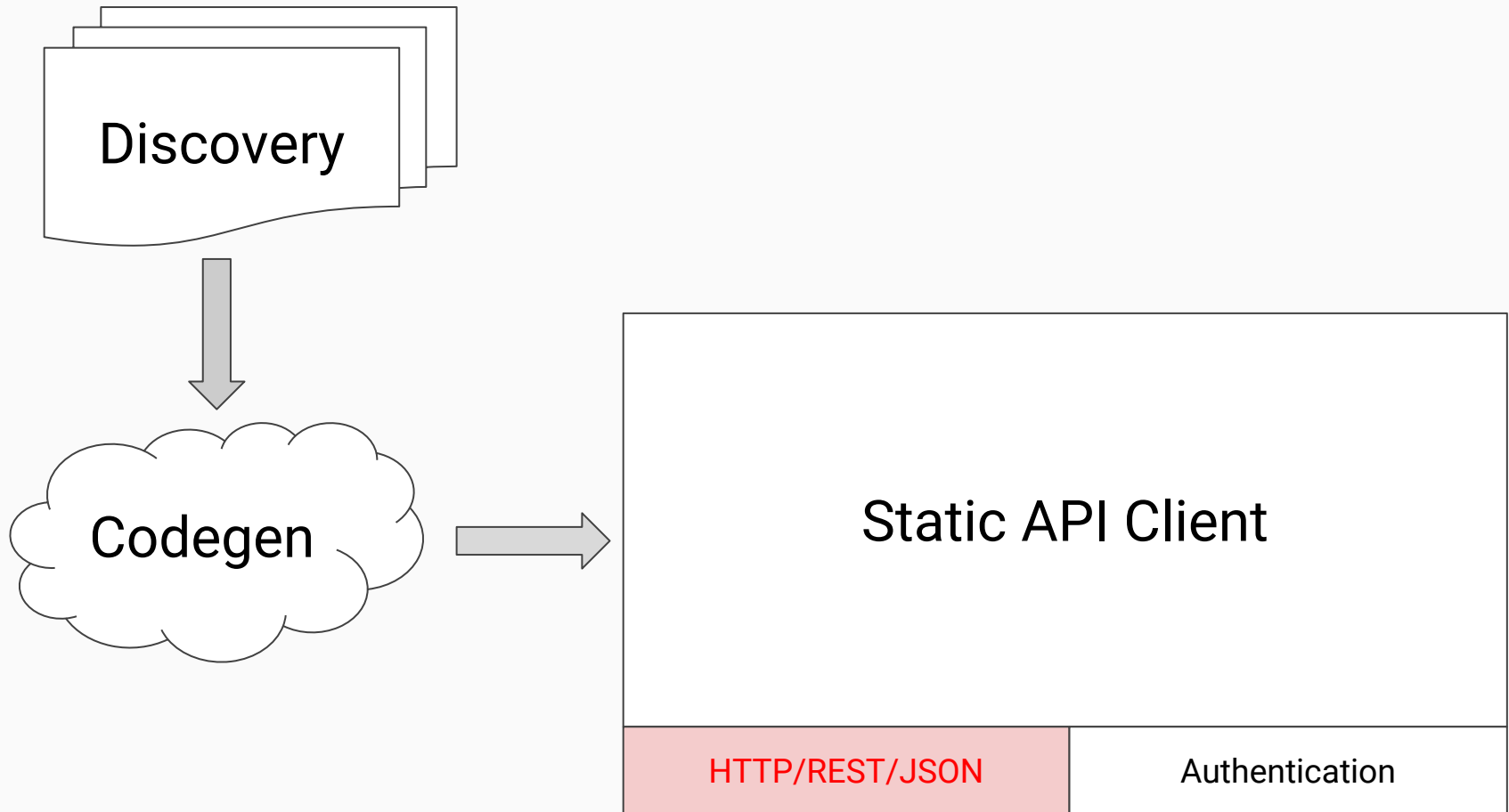
Results

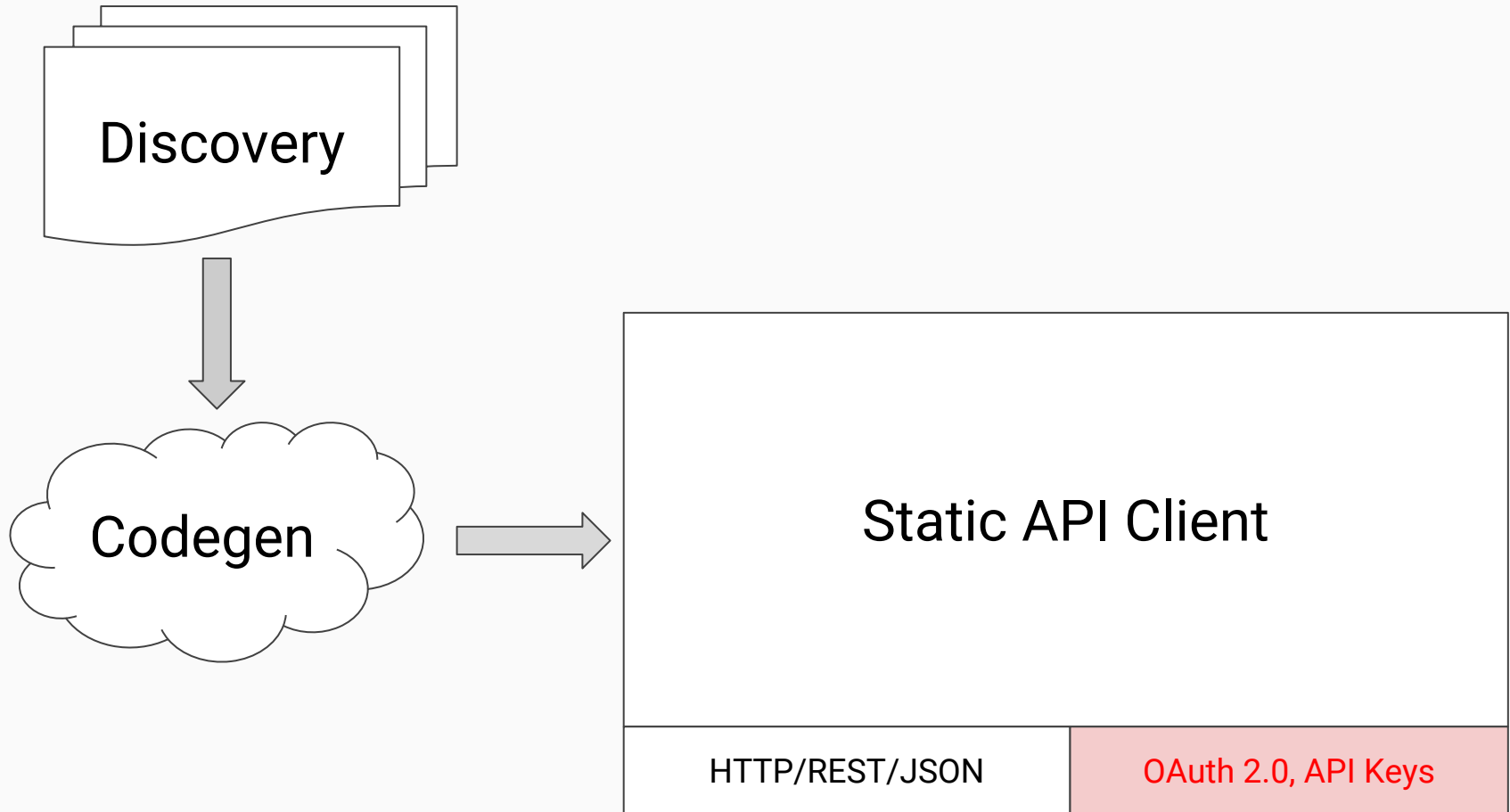
- Worked
- Not developer friendly
- Not transparent

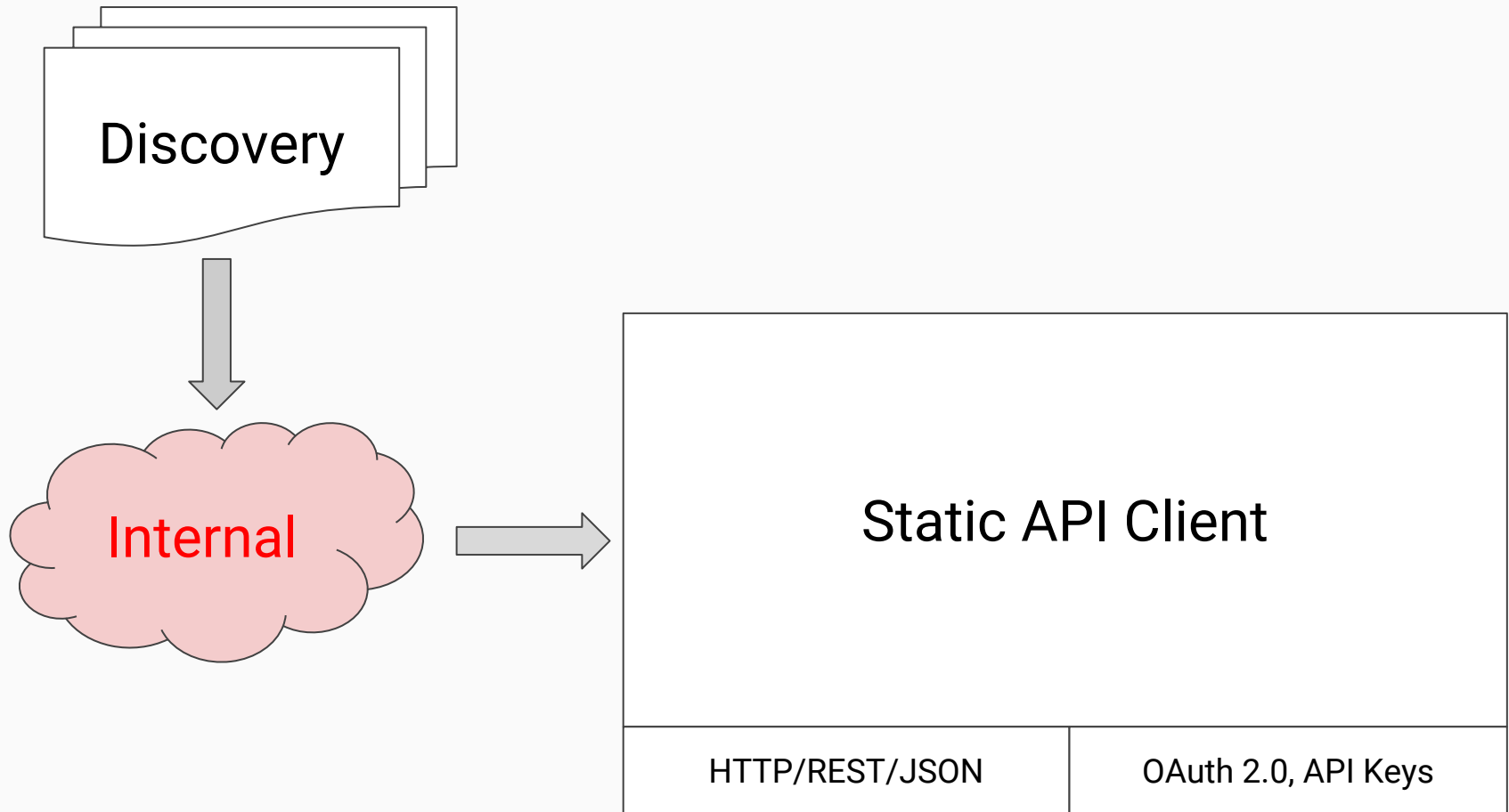
Case Study 2: Google APIs - Static Clients



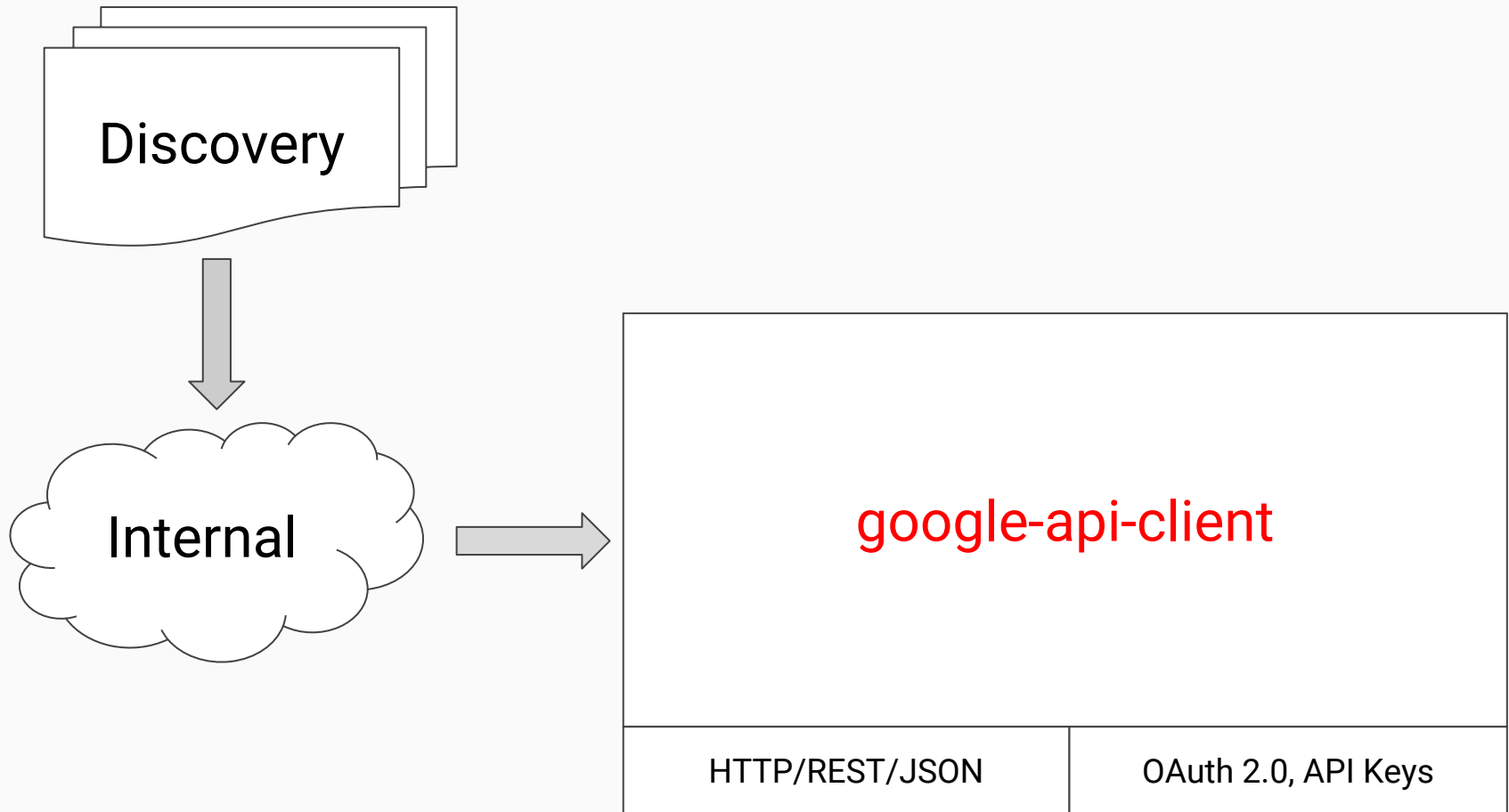








Google APIs - Static Clients



Results

- Works and scaled
- Closed source pipeline
- Not so developer friendly

Case Study 3: Cloud APIs on GCP

Goals

- Great developer experience

Goals

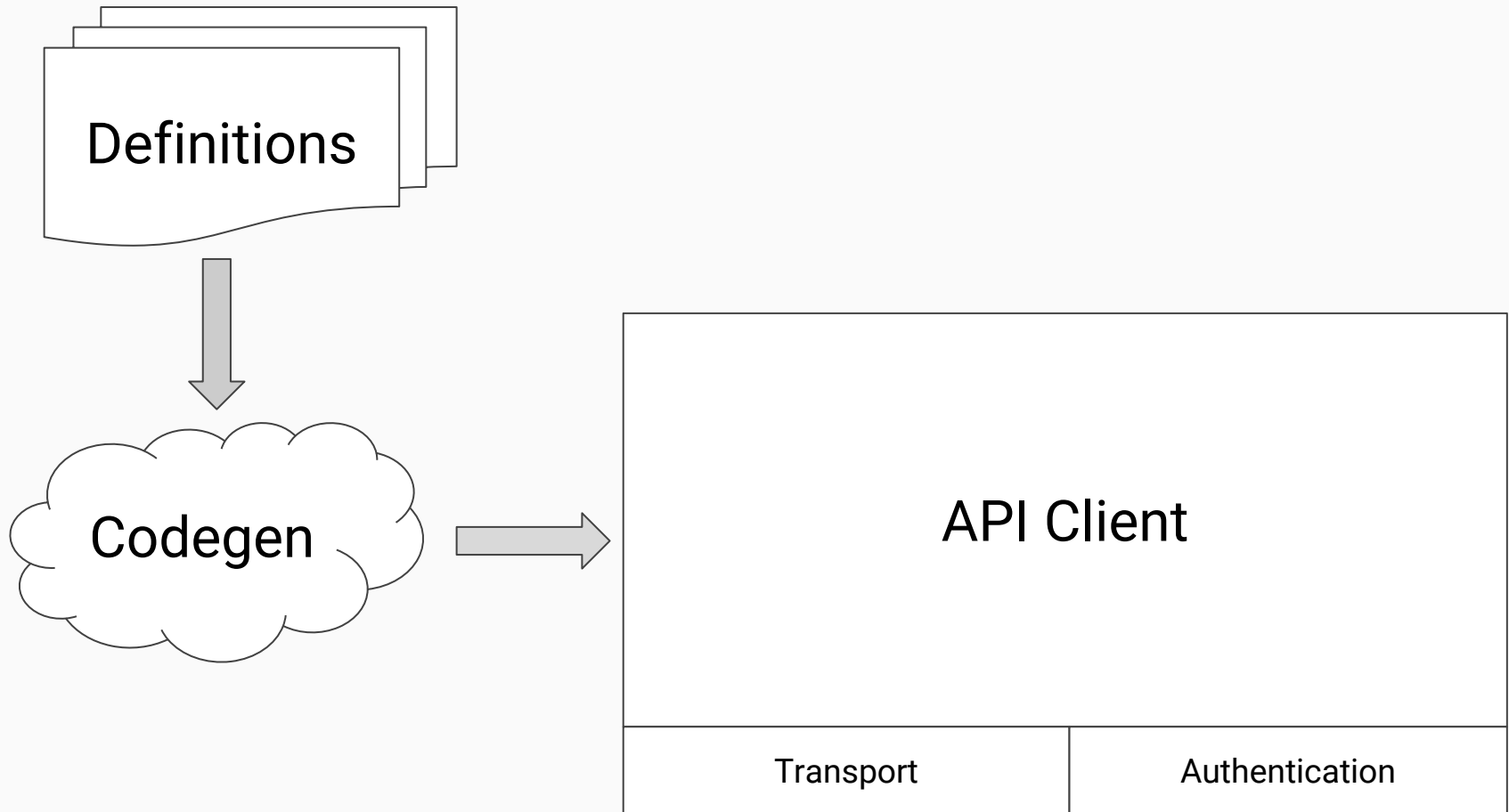
- Great developer experience
- All Cloud APIs work performantly

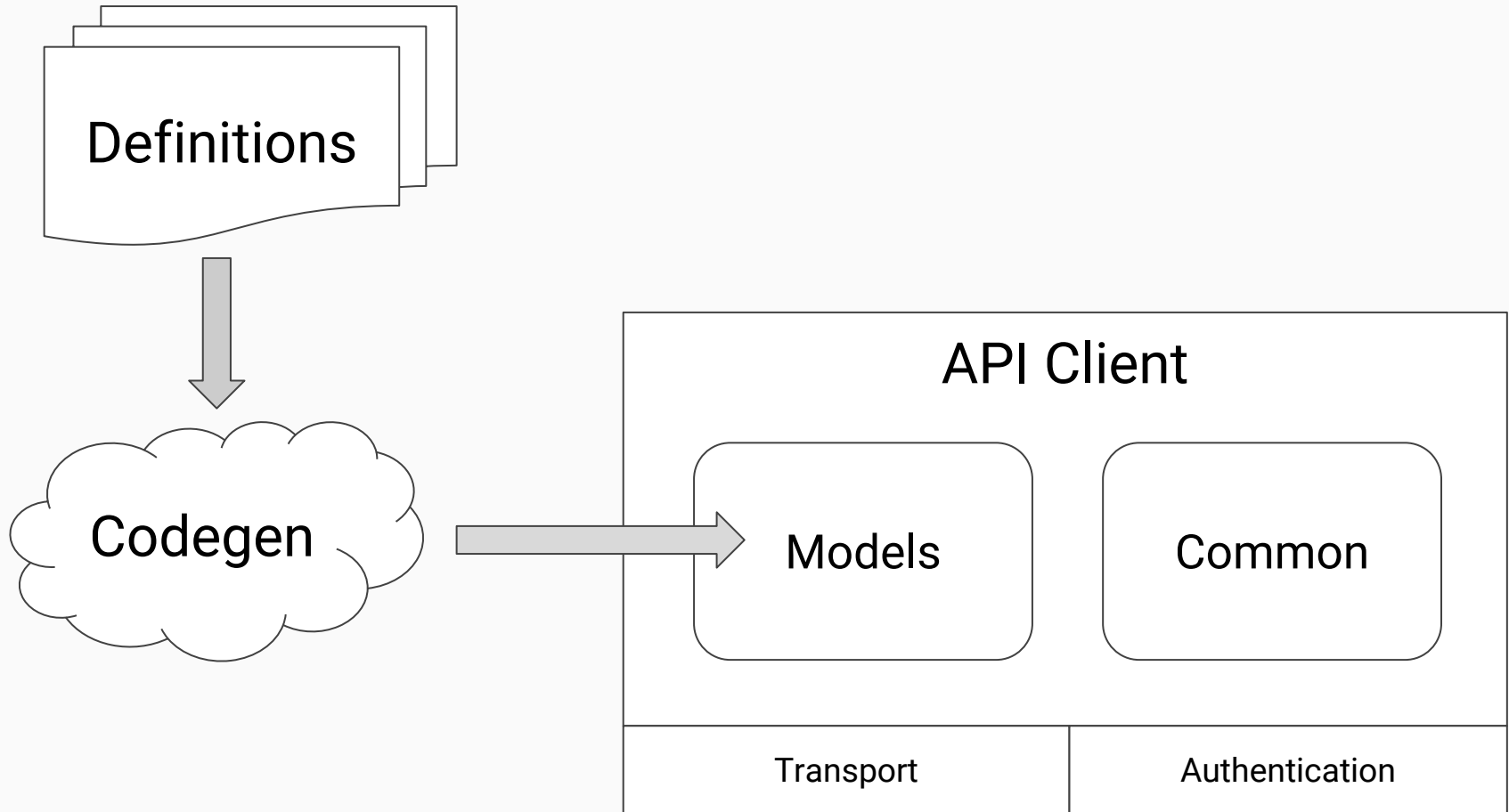
Goals

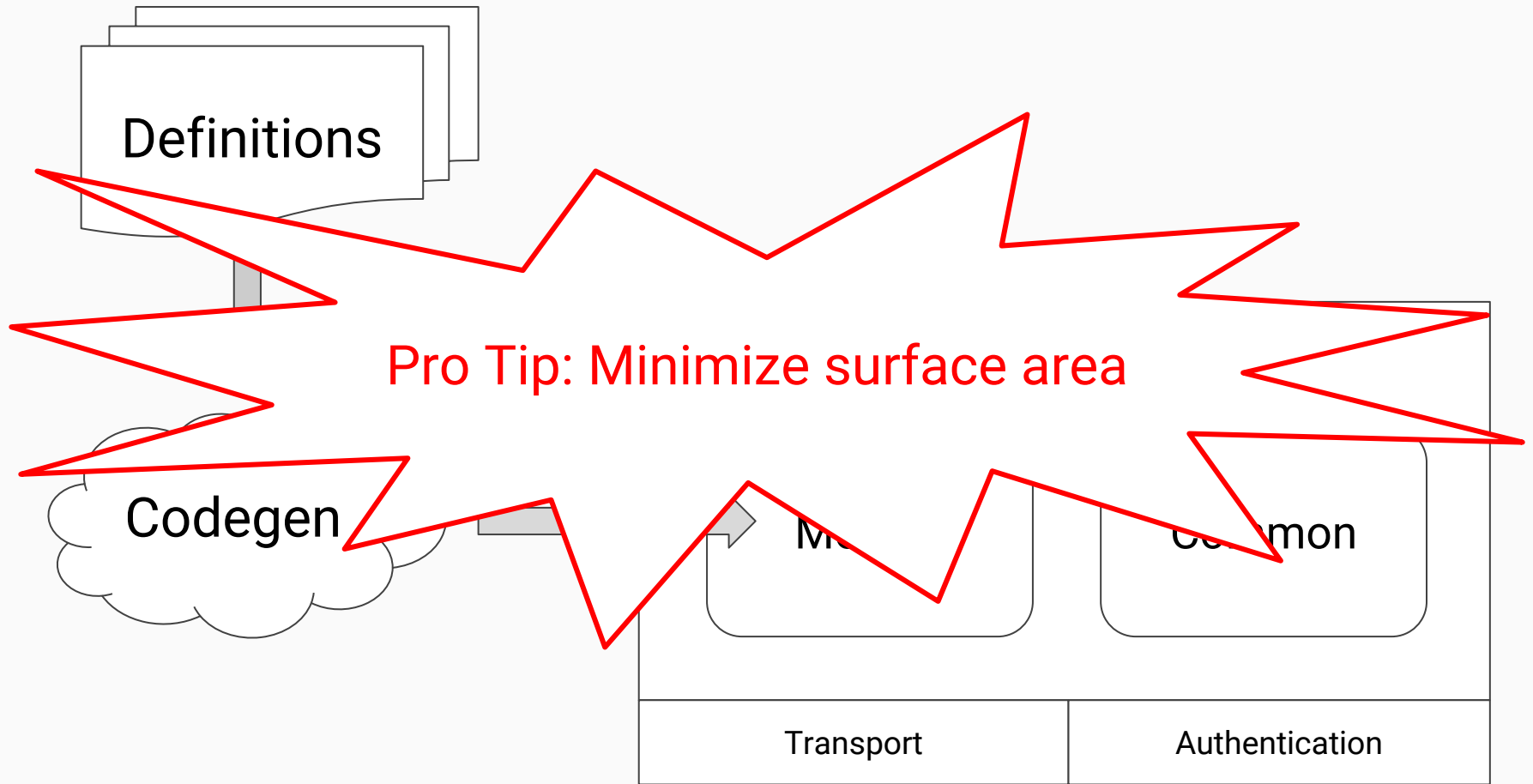
- Great developer experience
- All Cloud APIs work performantly
- Idiomatic

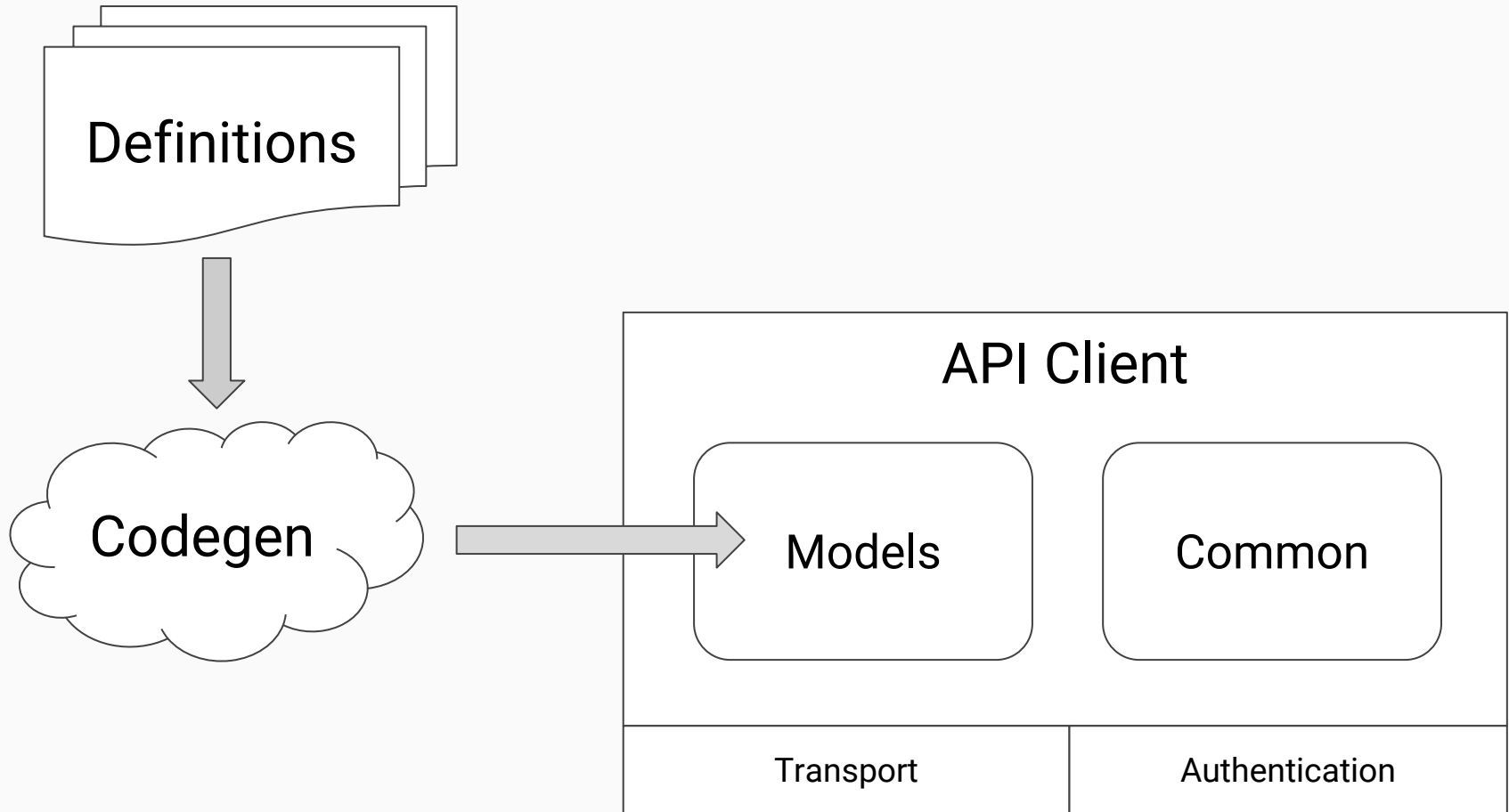
Goals

- **Great developer experience**
- All Cloud APIs work performantly
- Idiomatic

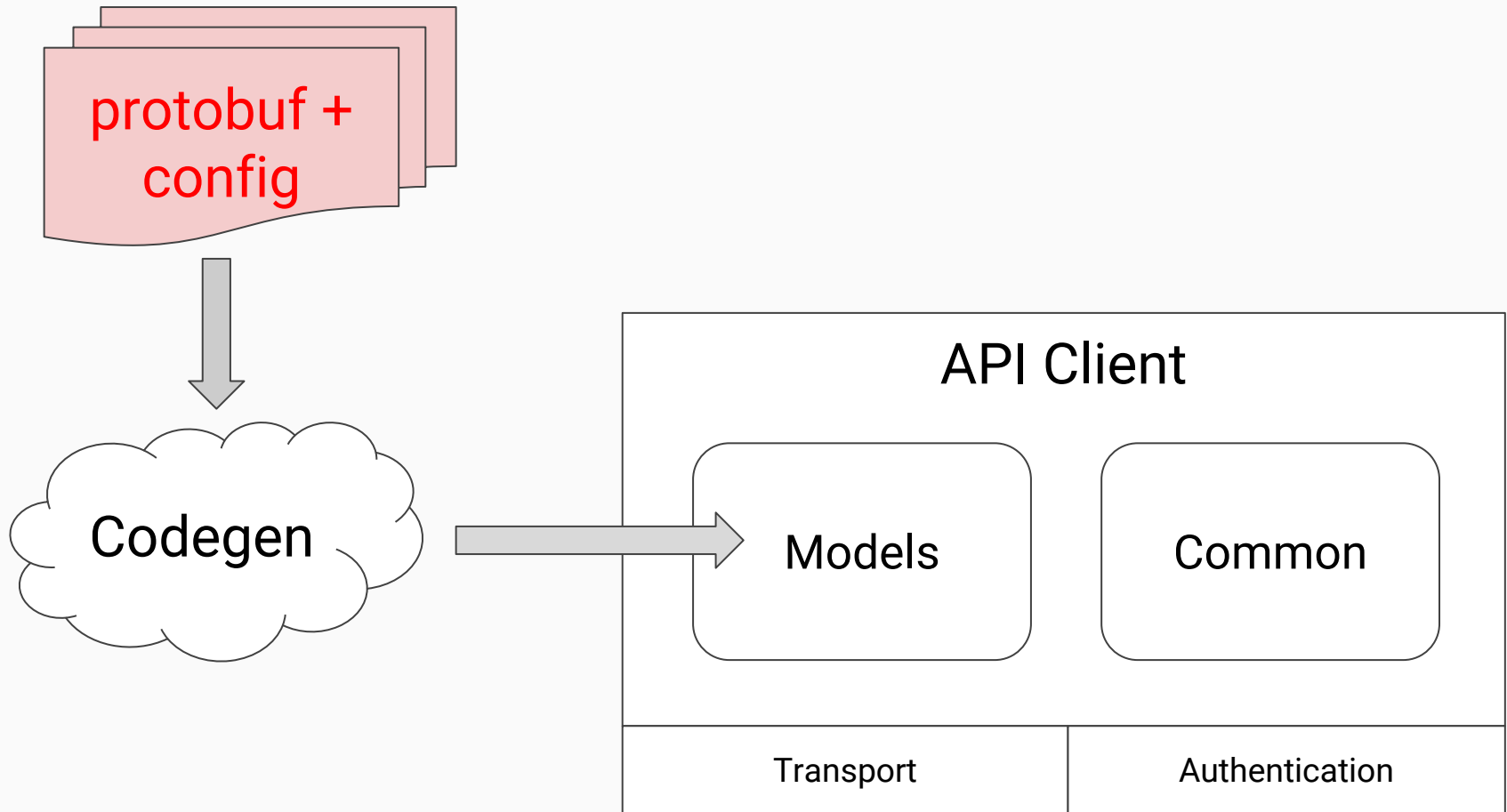








Cloud APIs on GCP



protobuf

```
message Person {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;

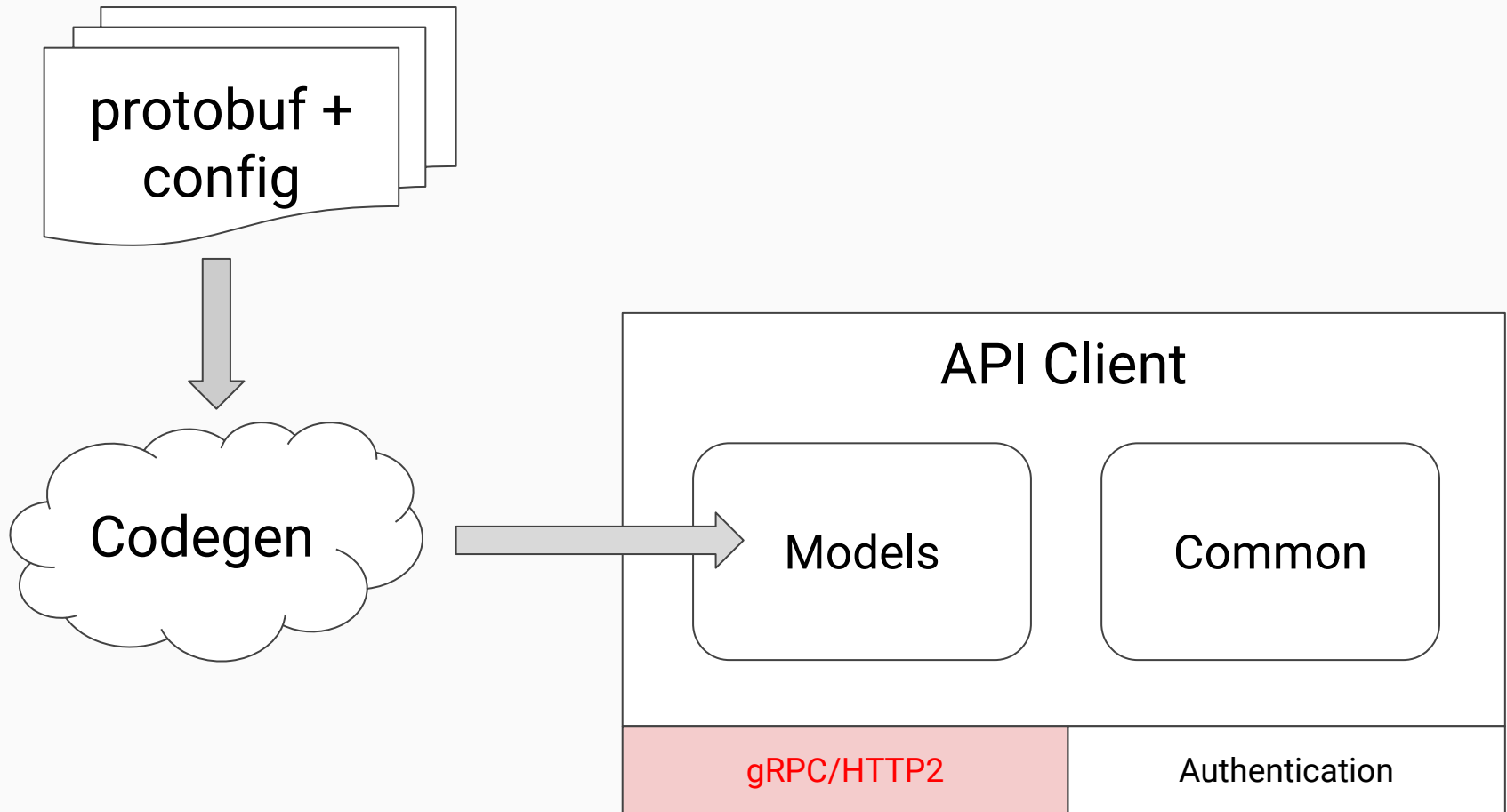
  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

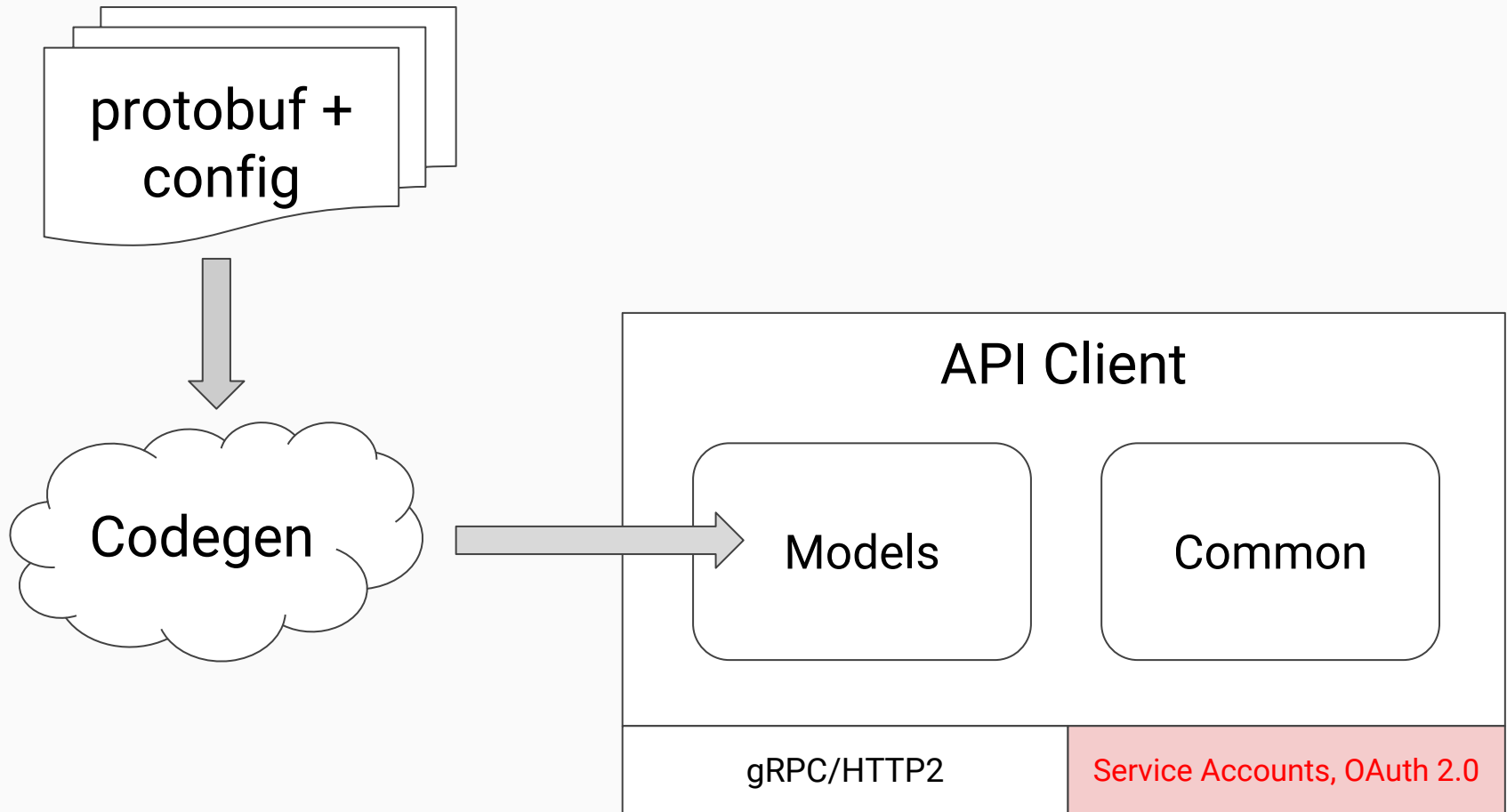
  message PhoneNumber {
    required string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phone = 4;
}

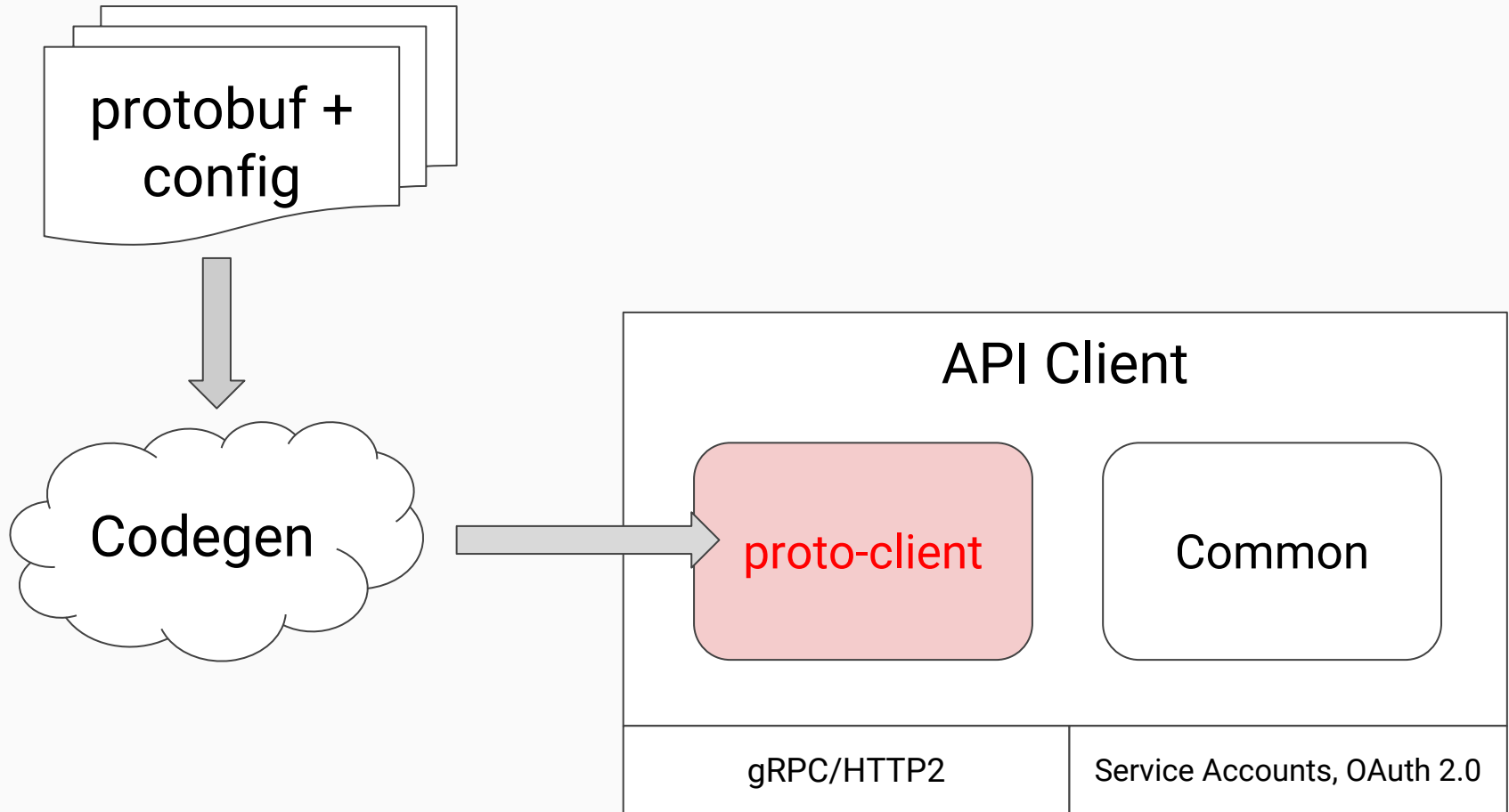
service SearchService {
  rpc Search (SearchRequest) returns (SearchResponse);
}
```

Cloud APIs on GCP

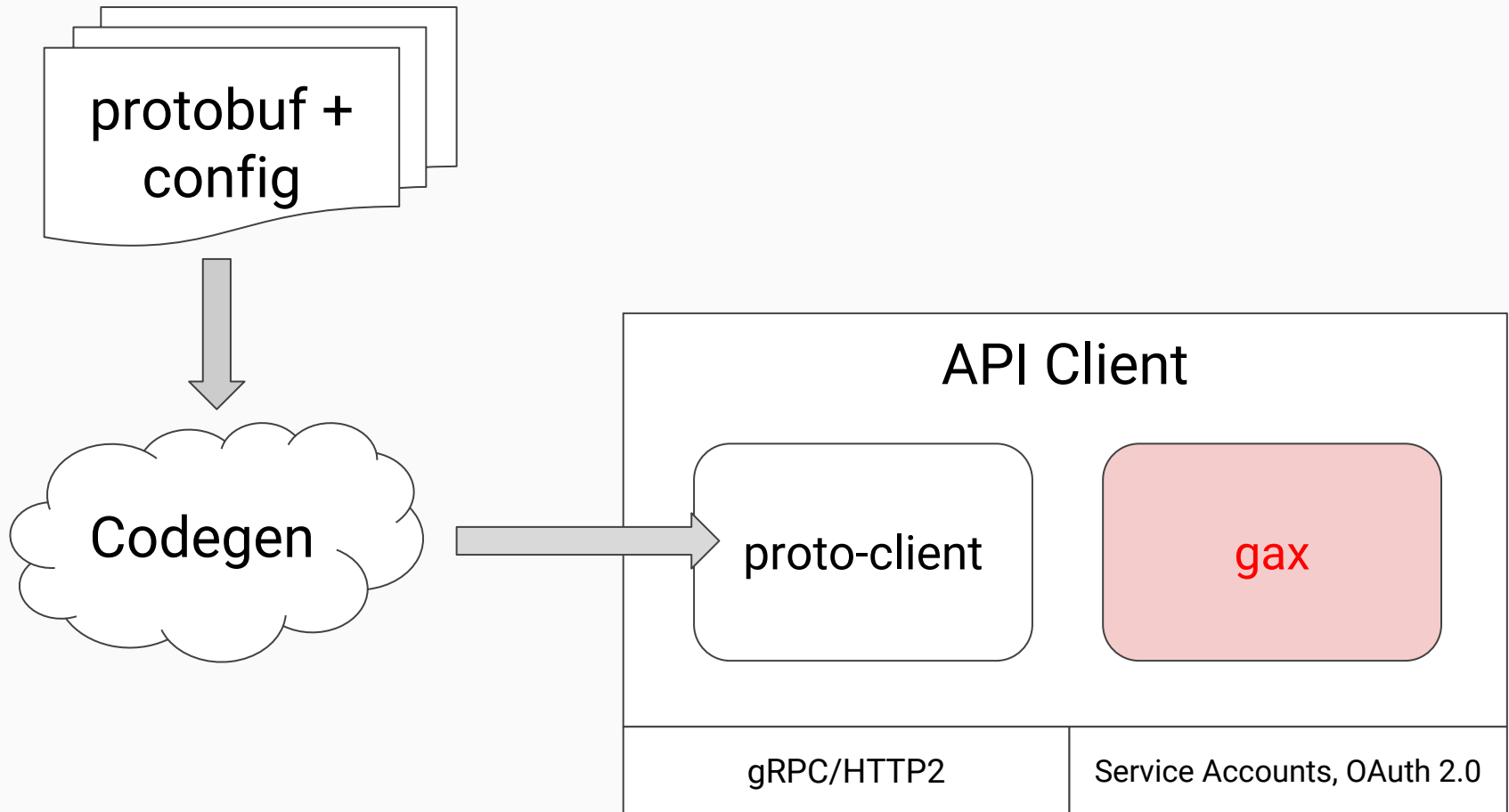


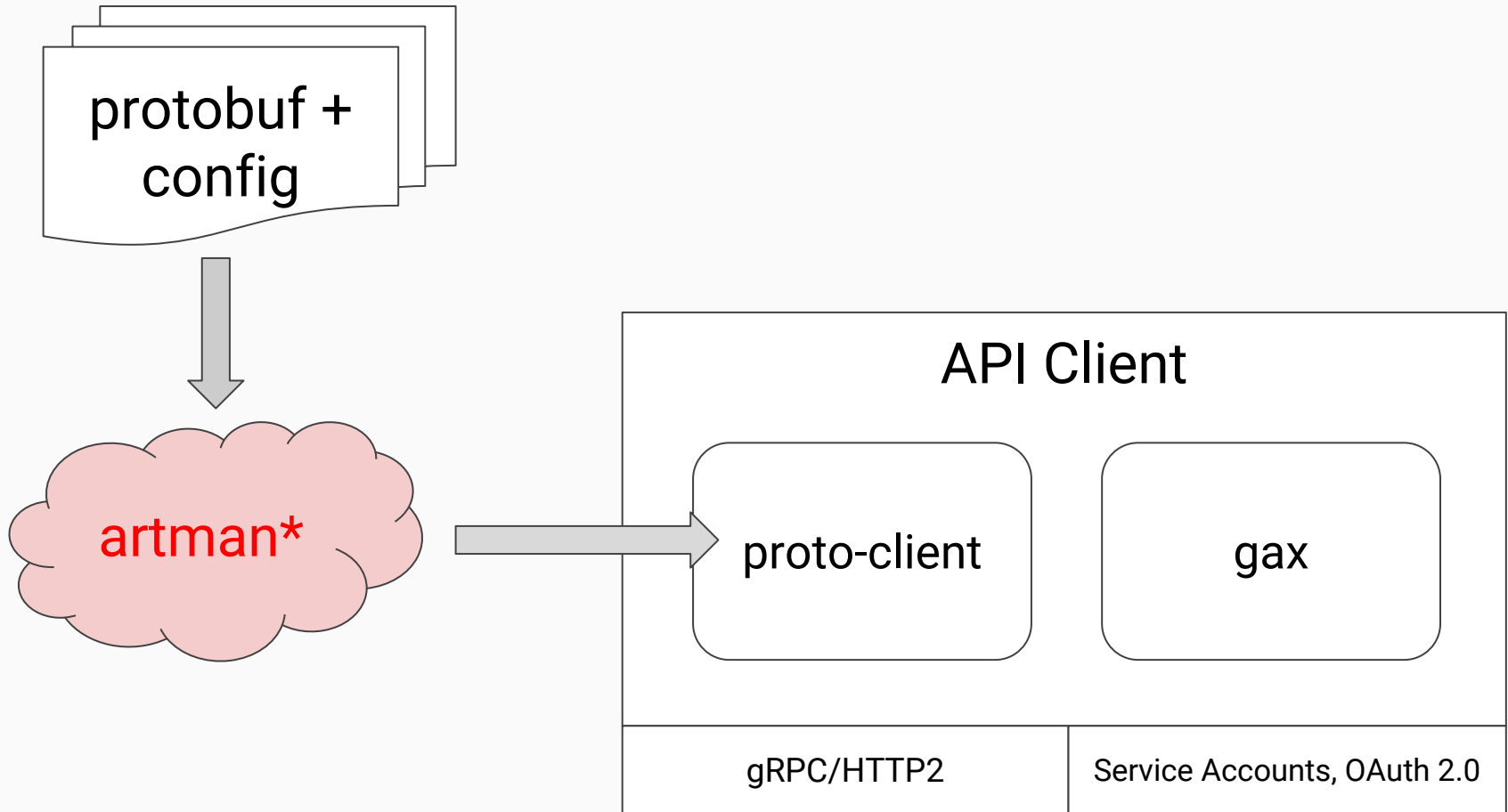


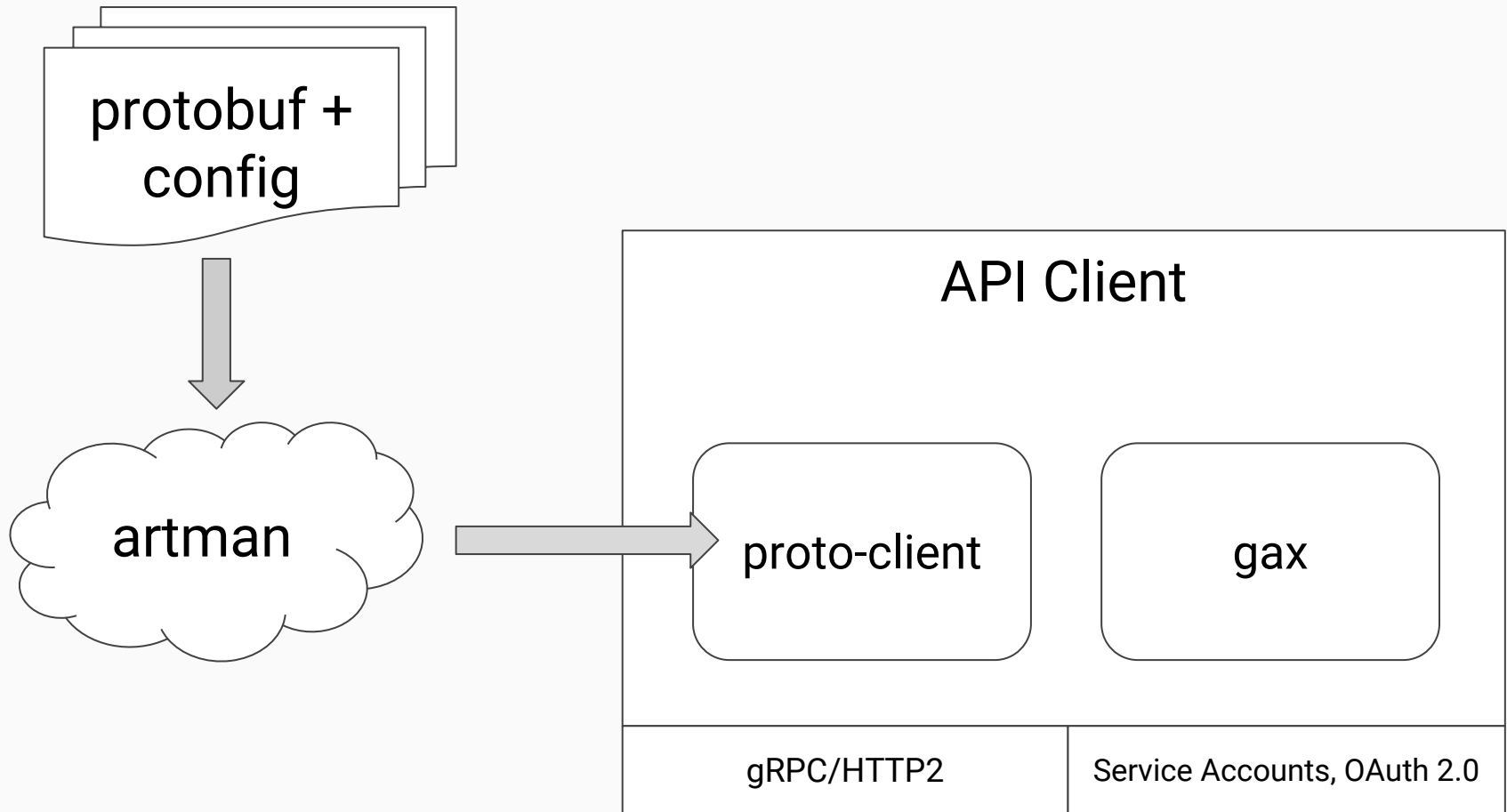
Cloud APIs on GCP

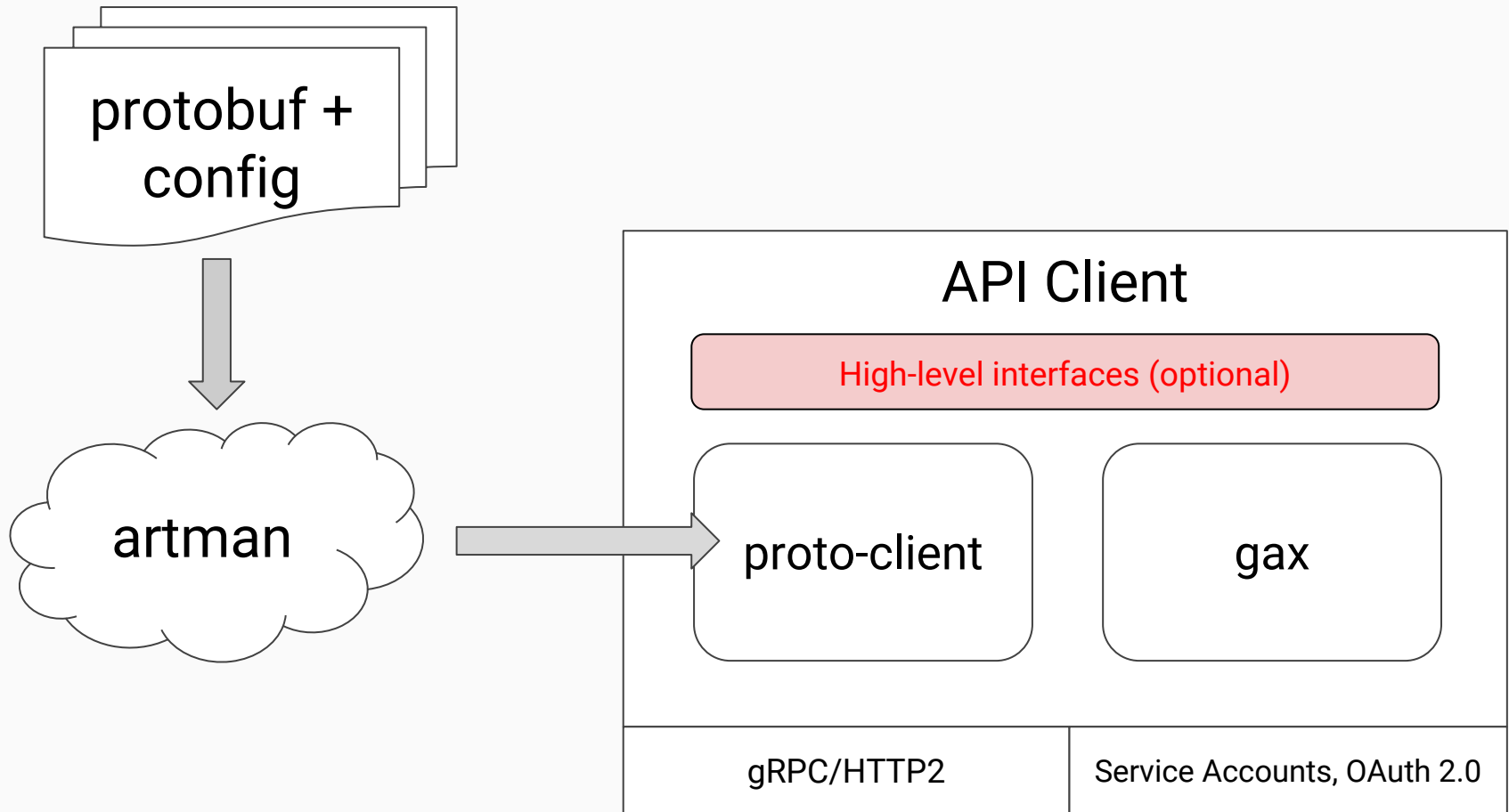


Cloud APIs on GCP

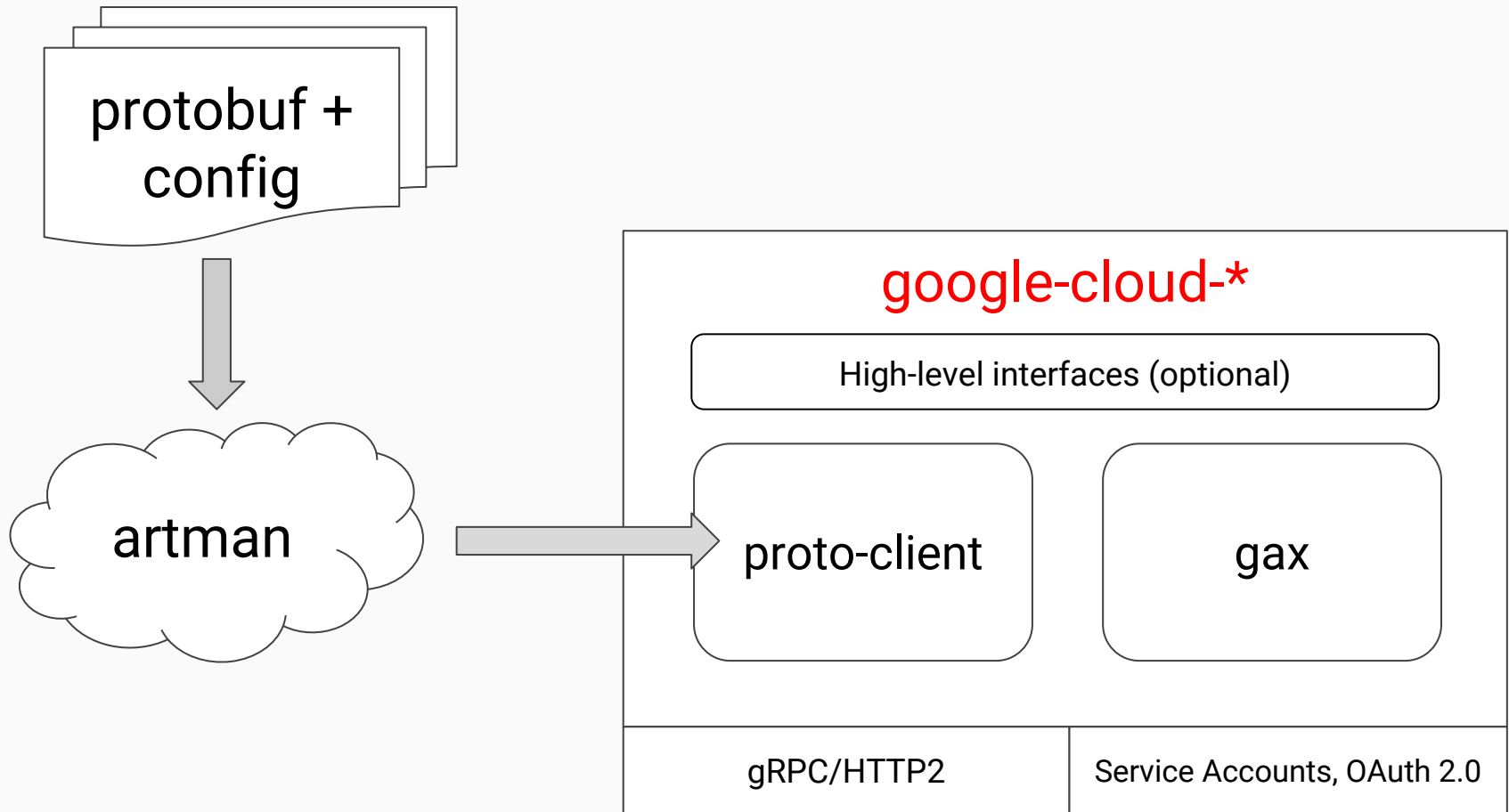








Cloud APIs on GCP



Results

- APIs work well
- Idiomatic
- Great developer experience

Challenges

- Closed source

Challenges

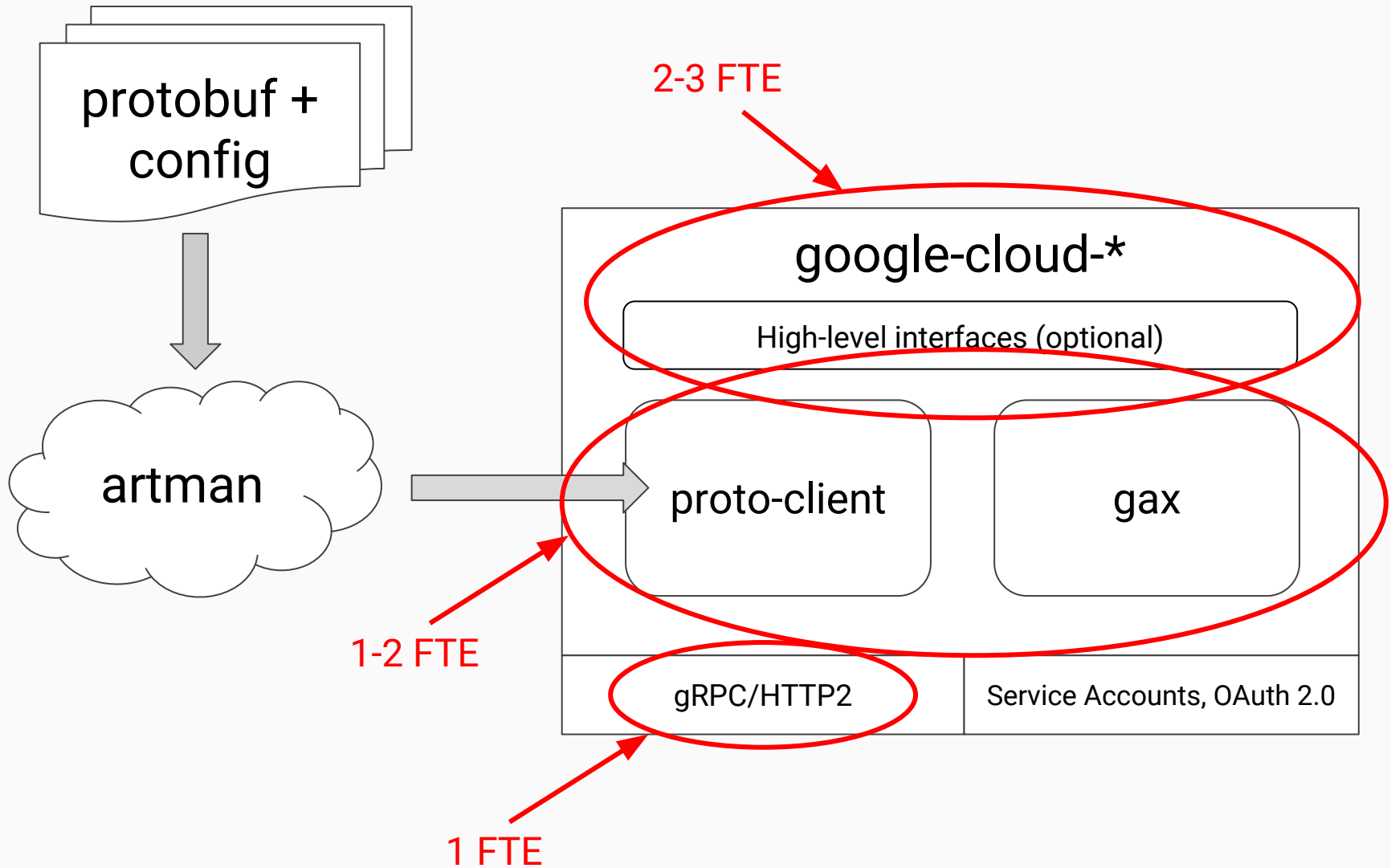
- Closed source
- Configuration format

Challenges

- Closed source
- Configuration format
- Modularity

Case Study 4: Elixir on GCP

Cloud APIs on GCP



Goals

- All APIs work

Goals

- All APIs work
- Good enough developer experience

Goals

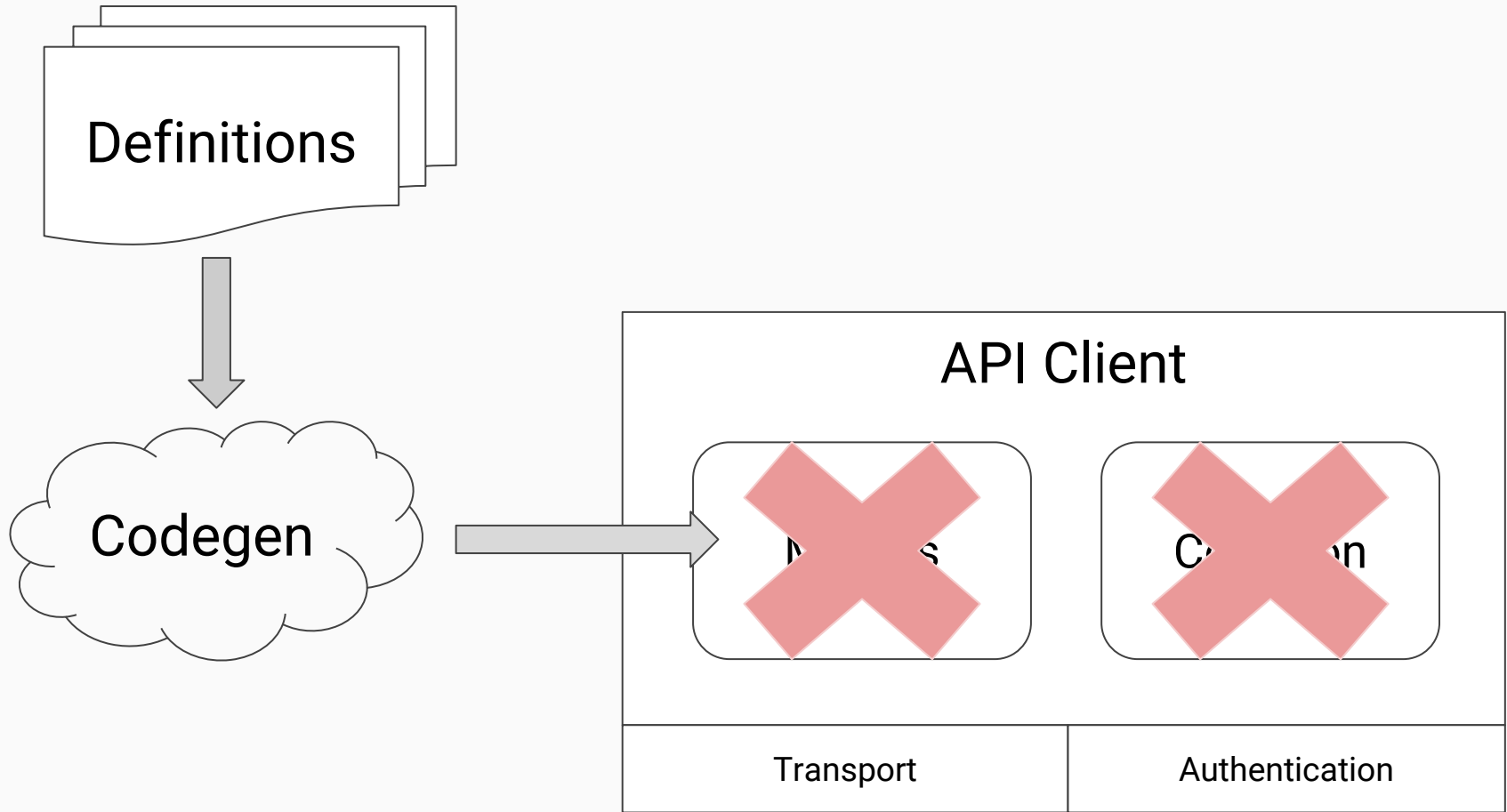
- All APIs work
- Good enough developer experience
- Demonstrate API client generation

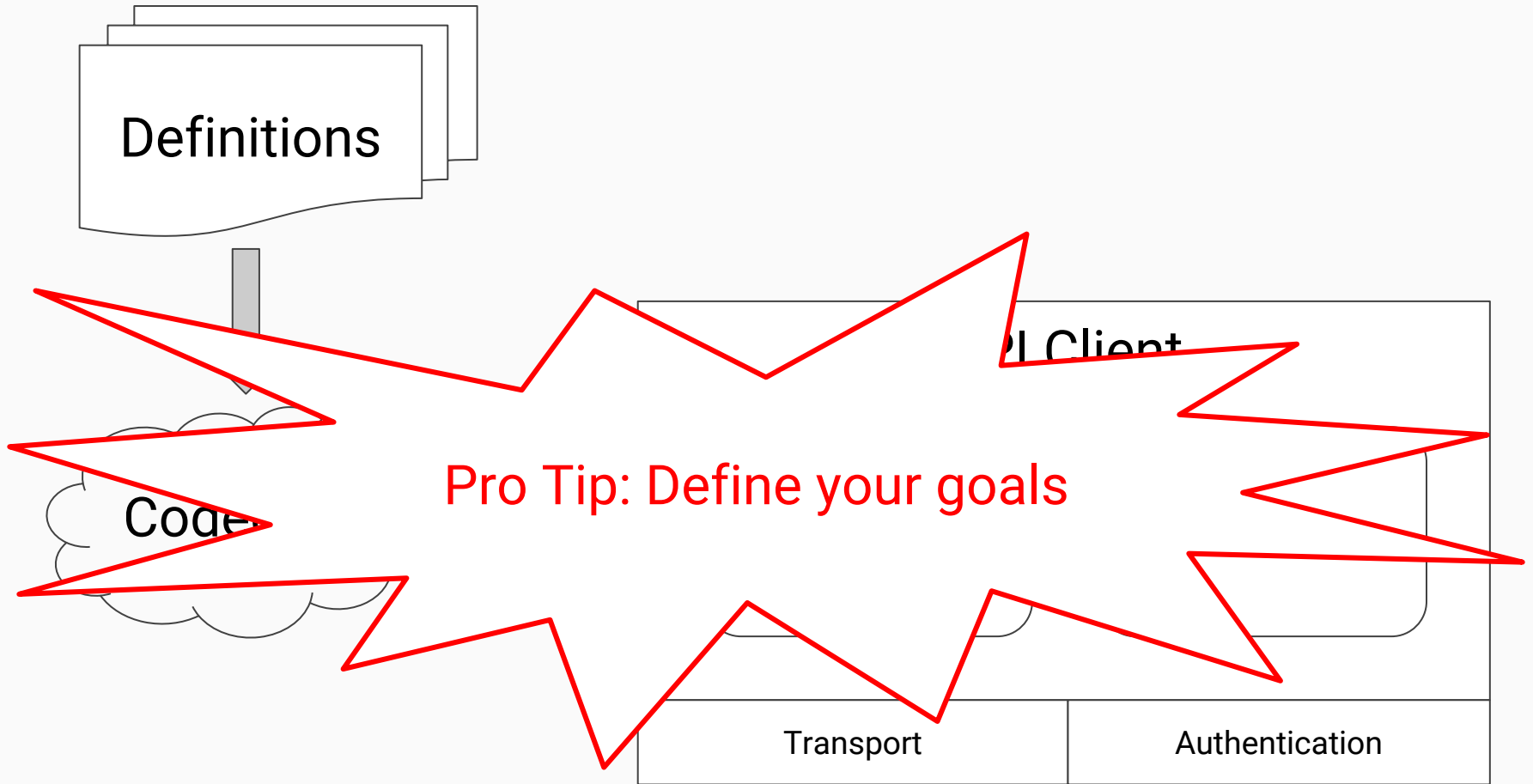
Goals

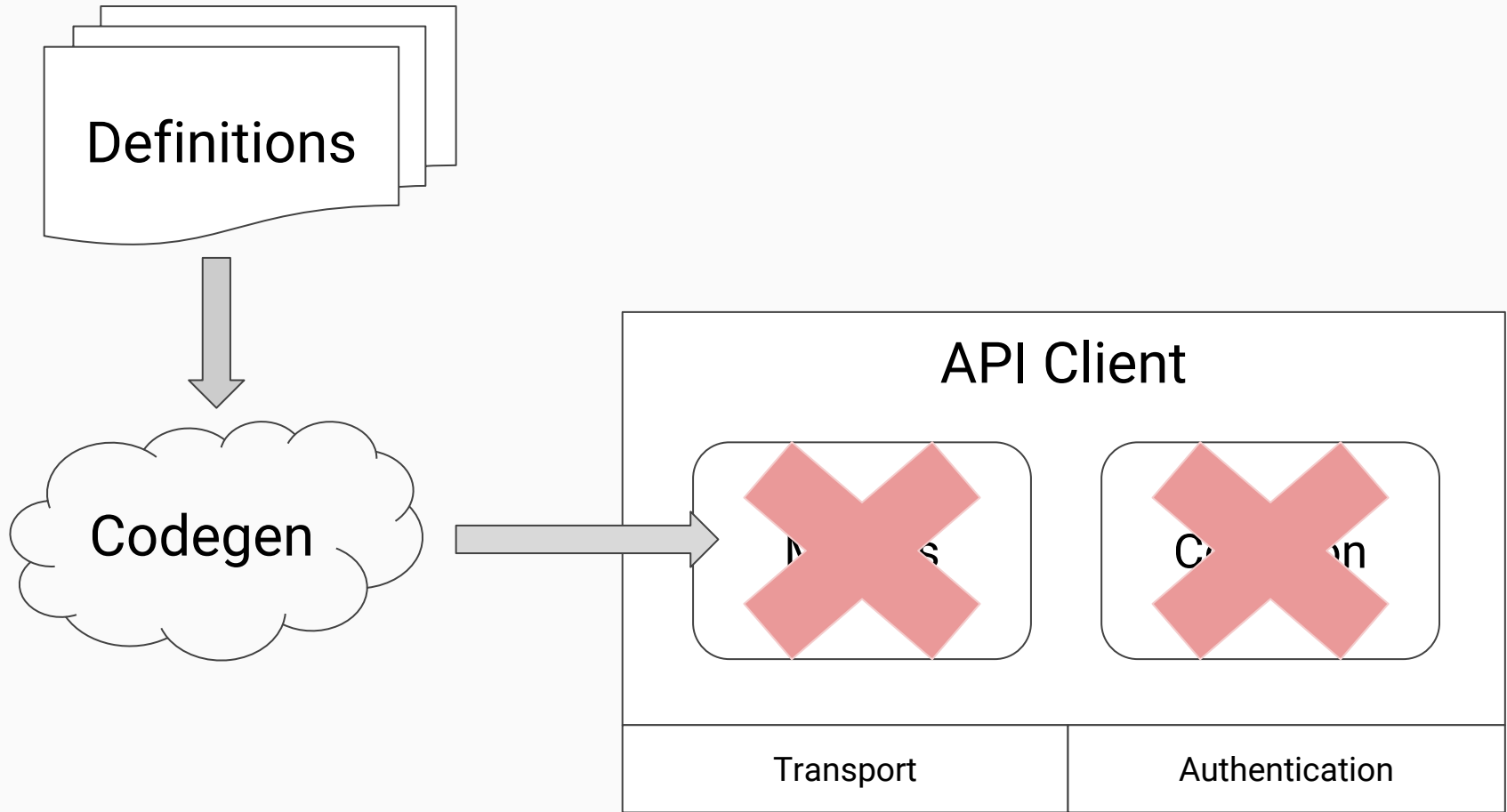
- All APIs work
- Good enough developer experience
- Demonstrate API client generation
- Path to full support

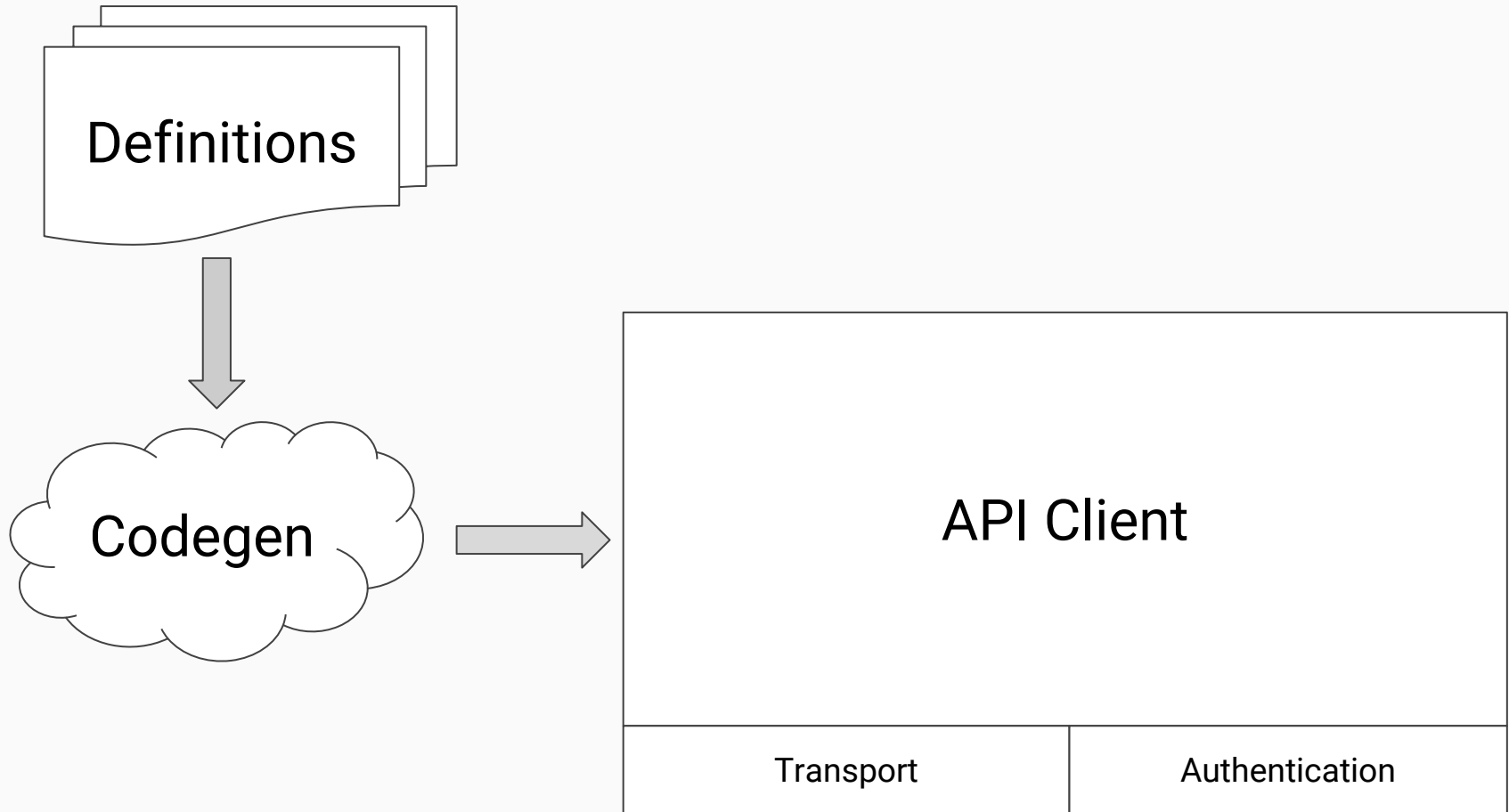
Goals

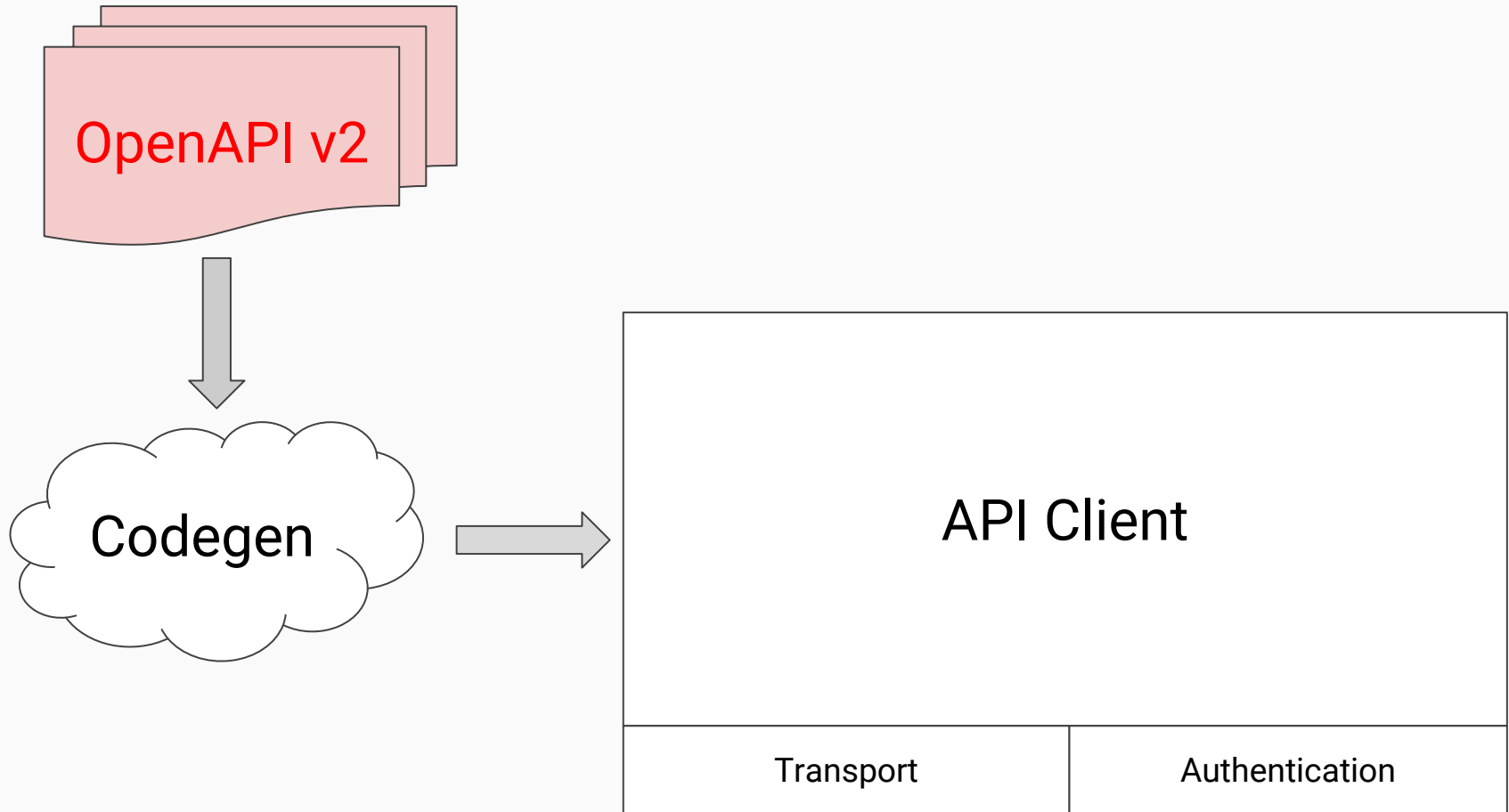
- All APIs work
- Good enough developer experience
- **Demonstrate API client generation**
- Path to full support







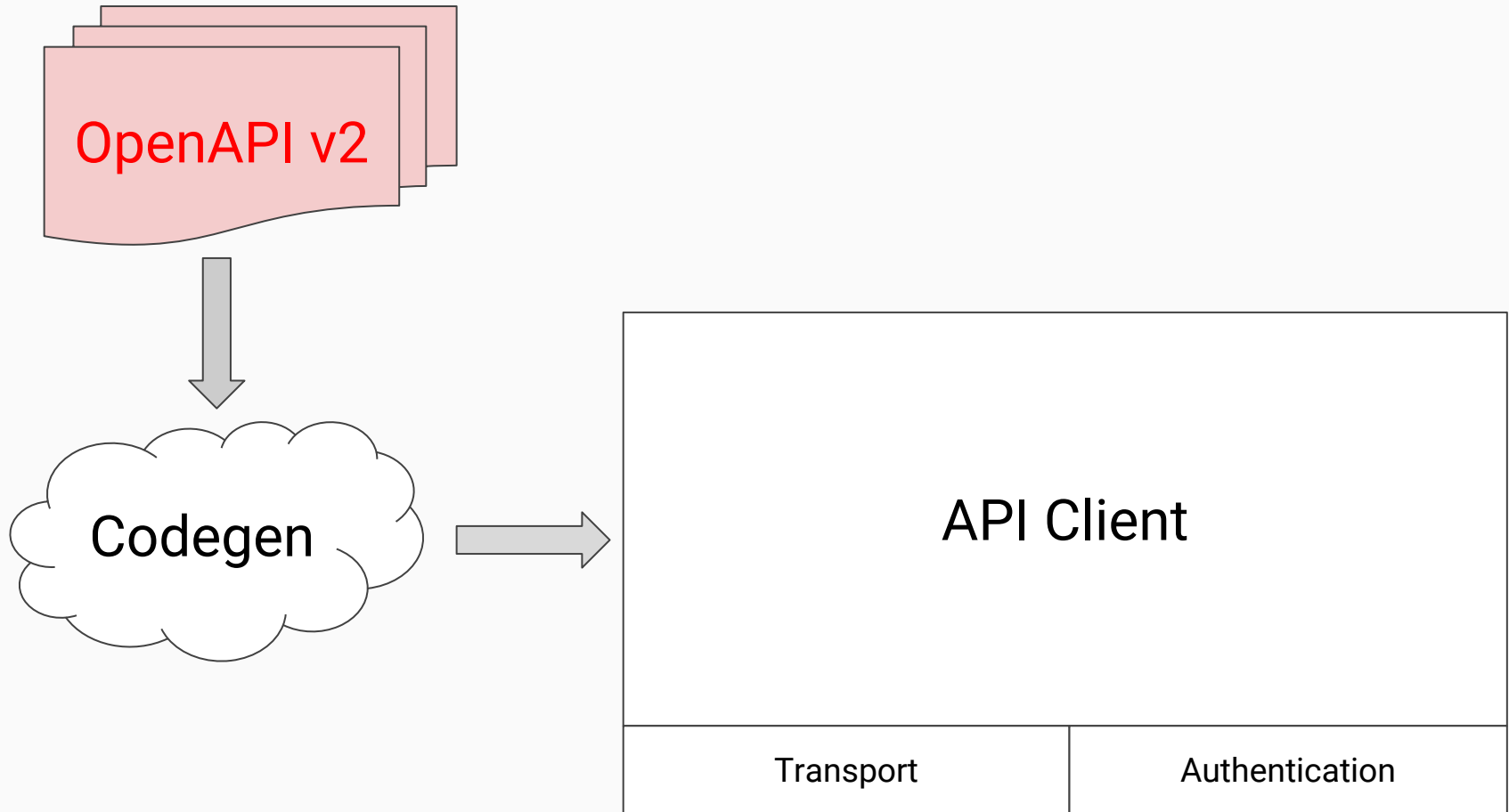


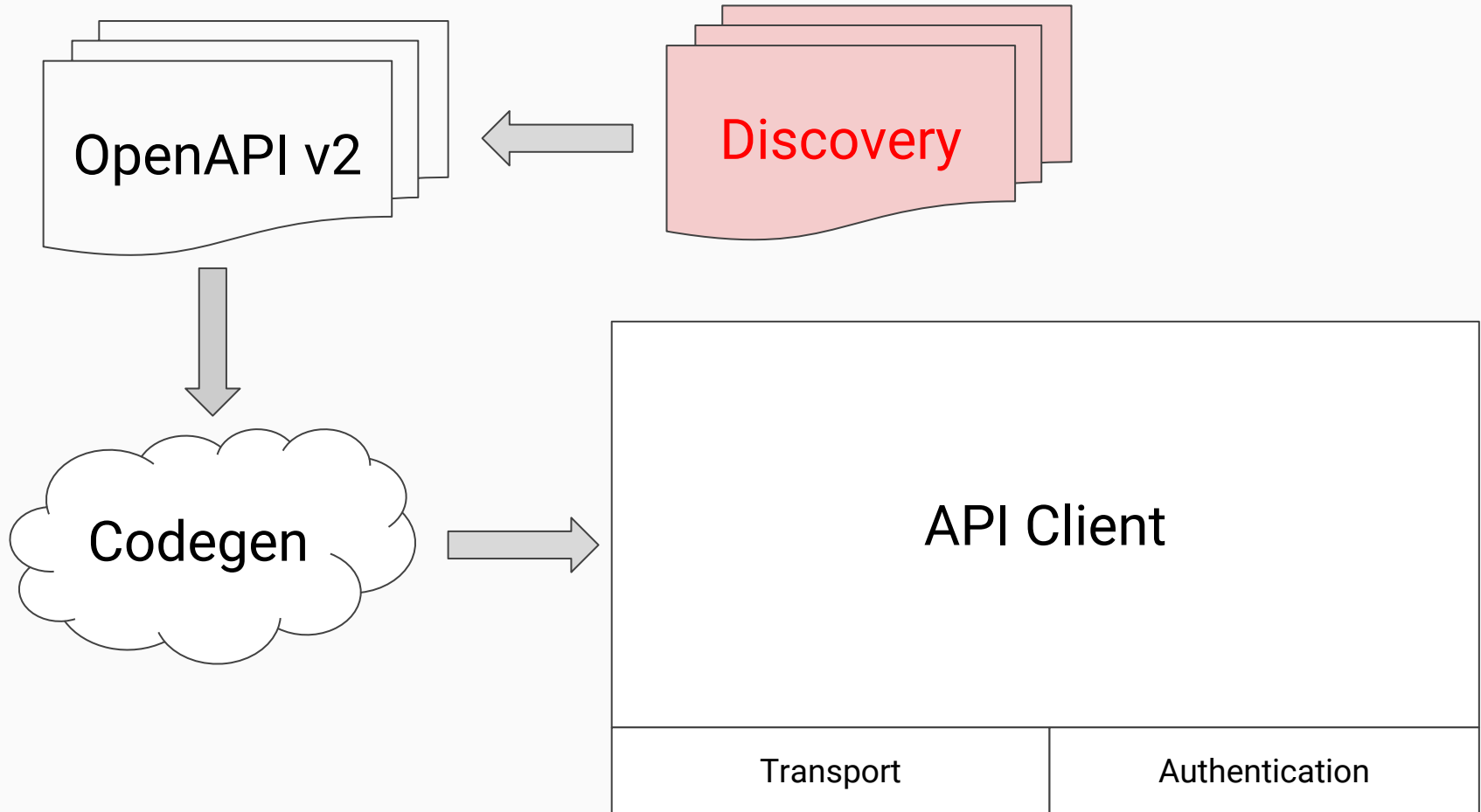


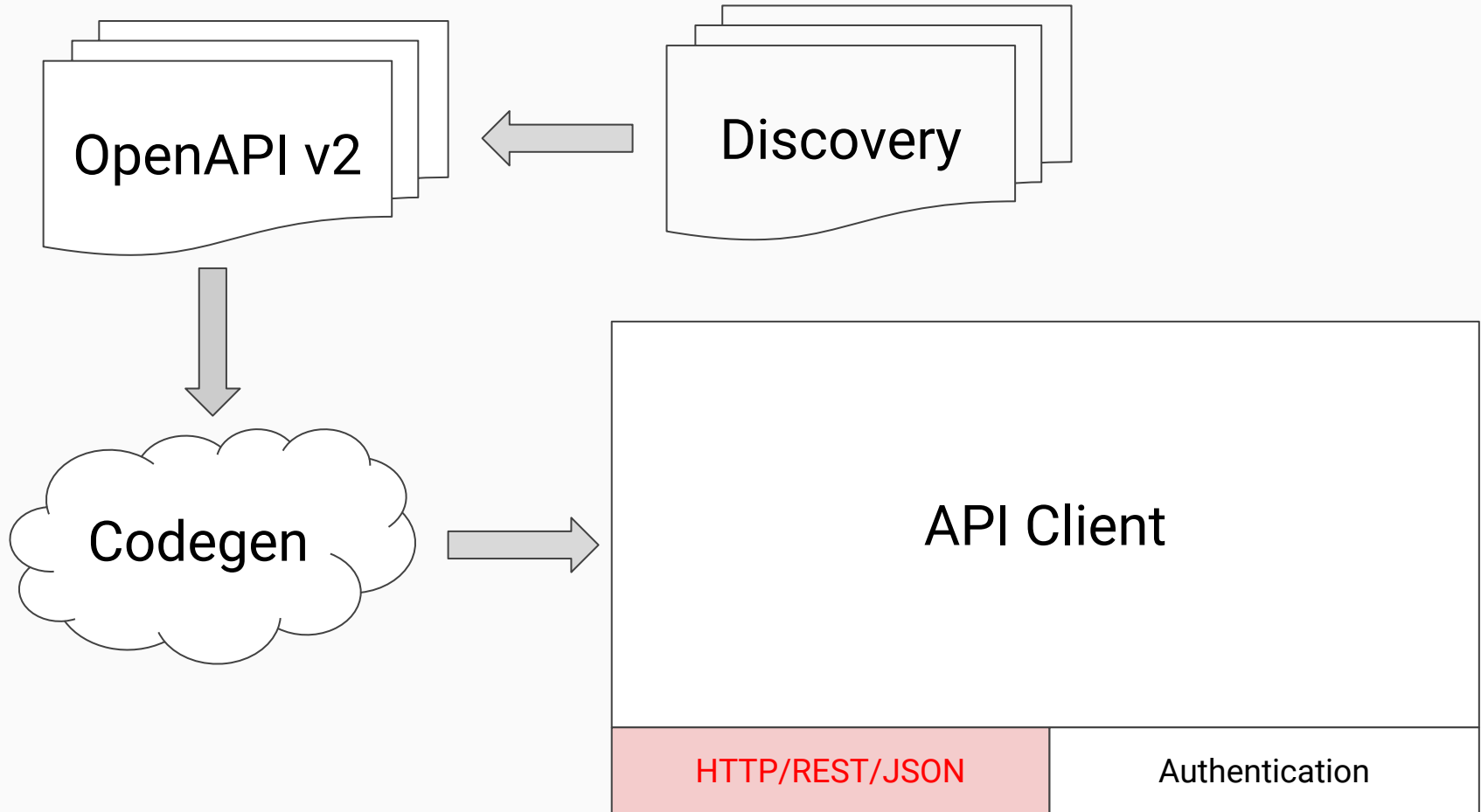
OpenAPI v2

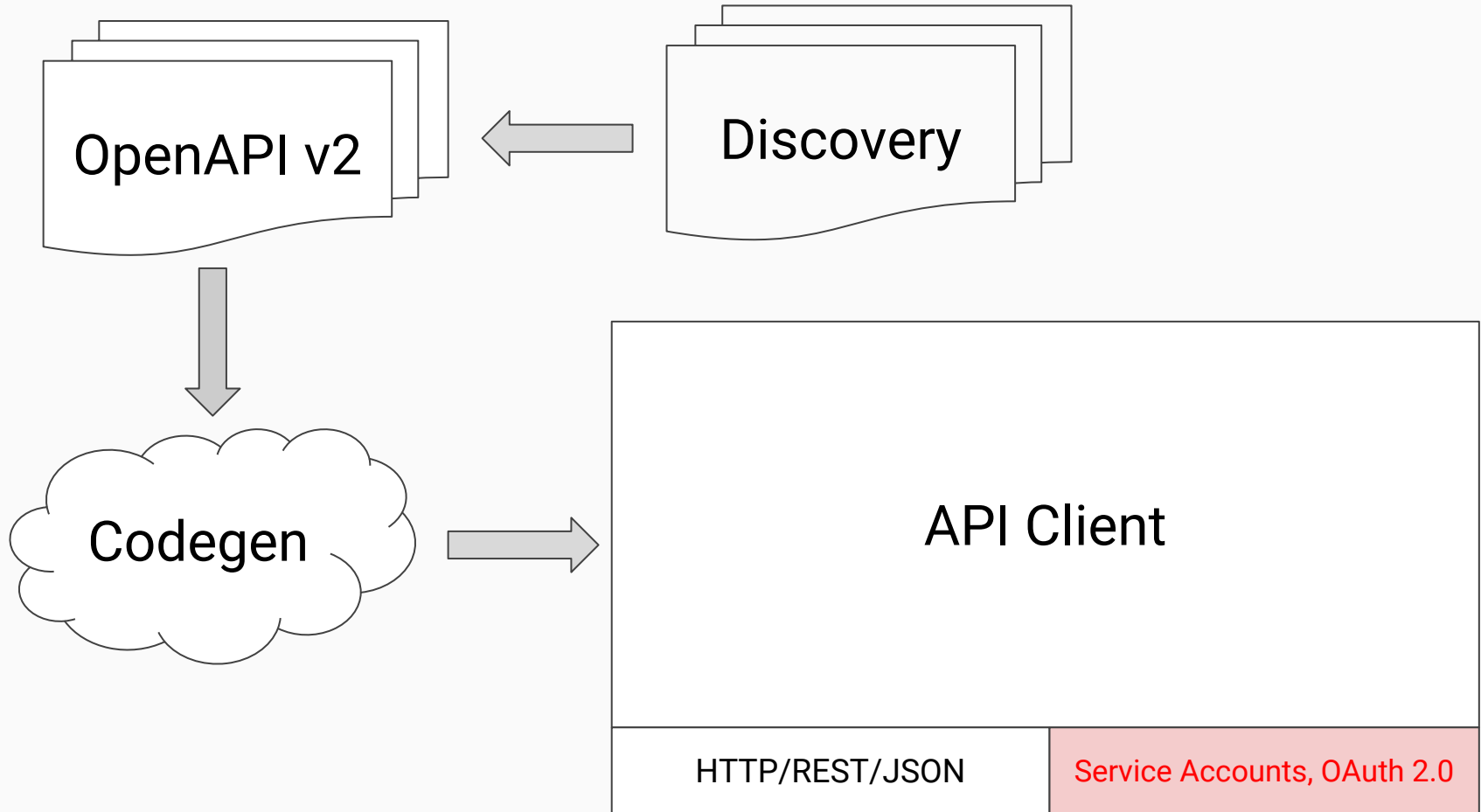
<https://www.openapis.org/>

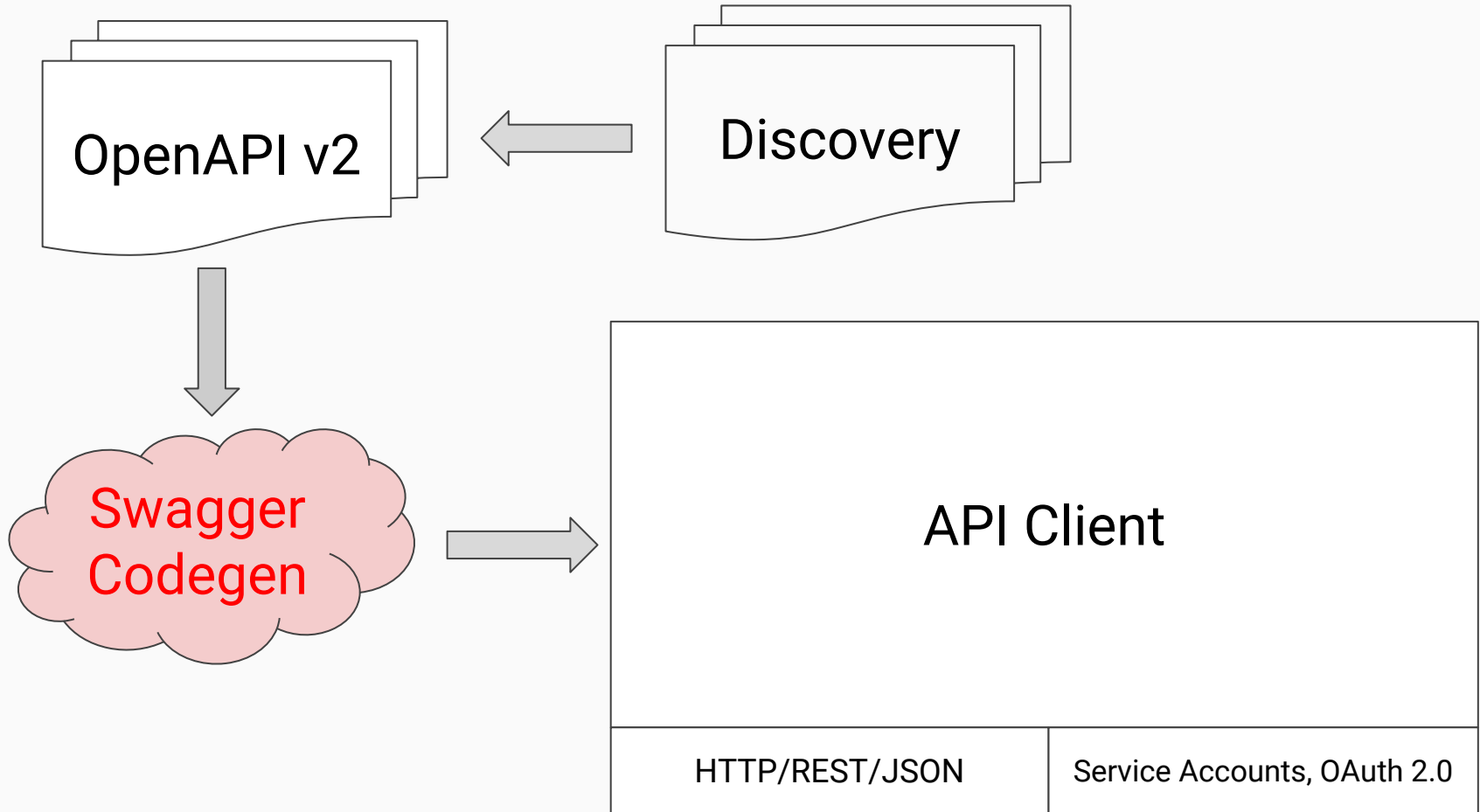
```
{
  "swagger": "2.0",
  "info": {
    "version": "1.0.0",
    "title": "Swagger Petstore",
    "license": {
      "name": "MIT"
    }
  },
  "host": "petstore.swagger.io",
  "basePath": "/v1",
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/pets": {
      "get": {
        "summary": "List all pets",
        "operationId": "listPets",
        "tags": [
          "pets"
        ],
      },
    },
  },
}
```

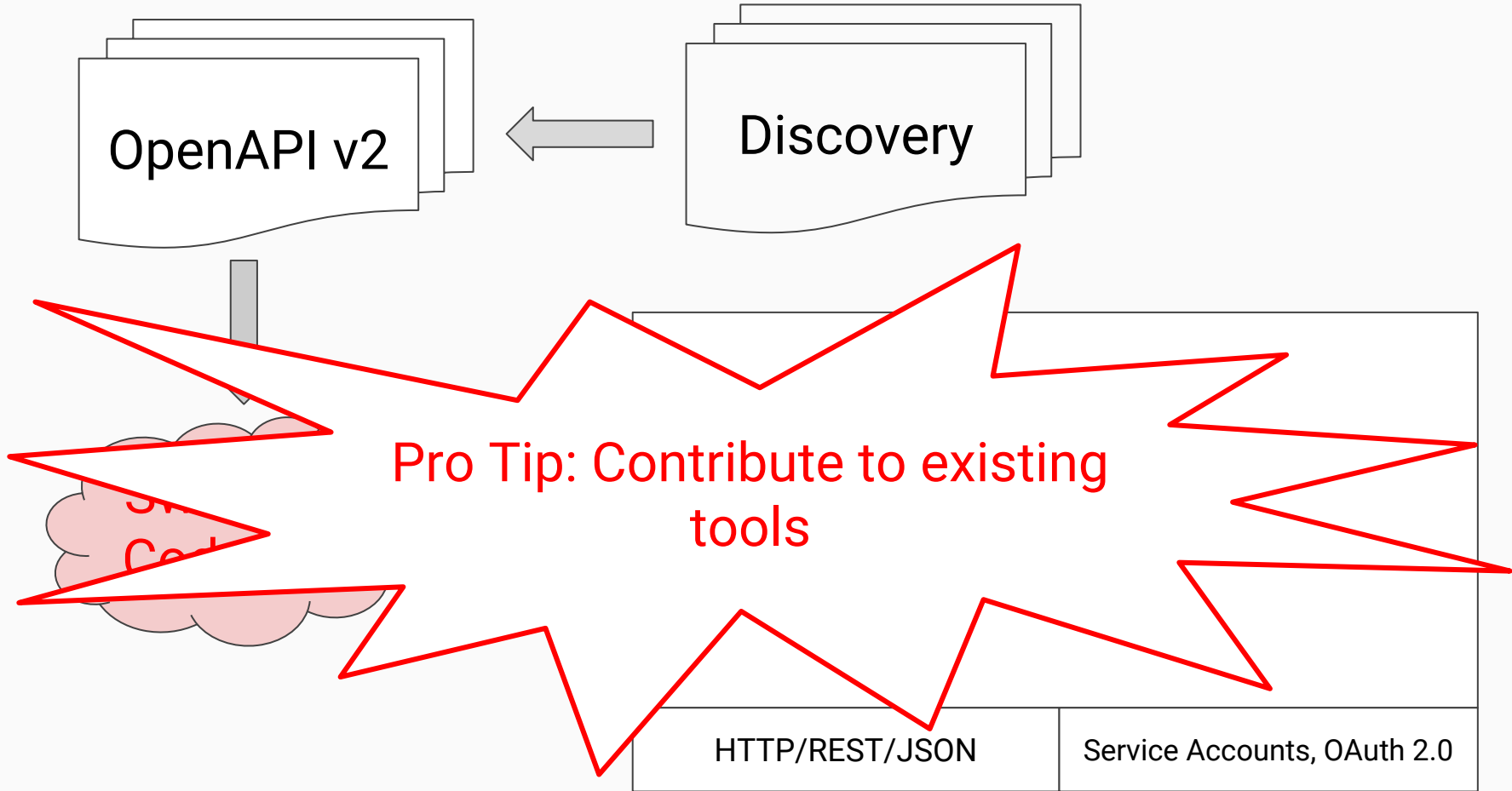


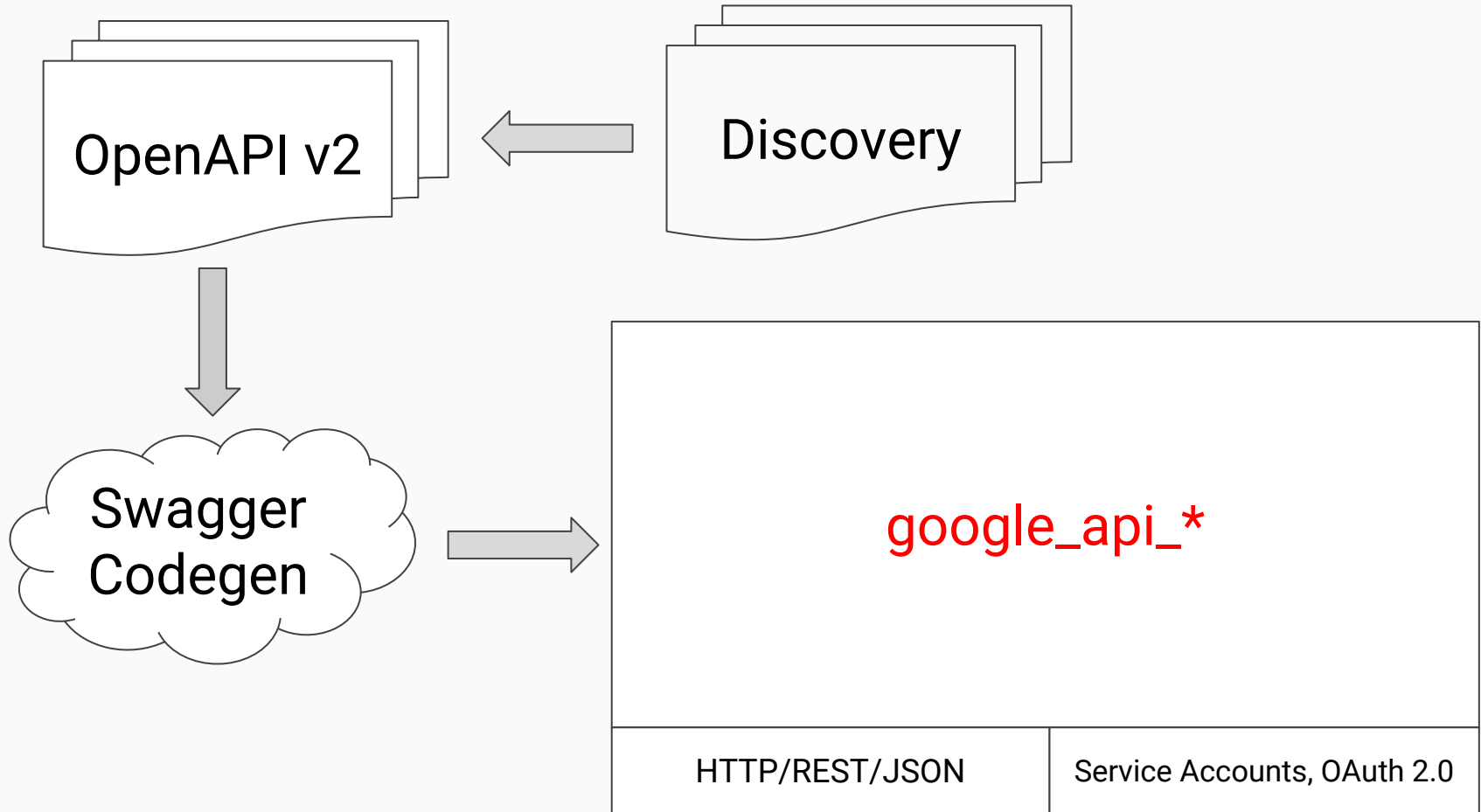












google-cloud

Owned packages

[google_api_accelerated_mobile_page_url](#) 0.0.1
[google_api_ad_exchange_buyer](#) 0.0.1
[google_api_ad_exchange_seller](#) 0.0.1
[google_api_ad_experience_report](#) 0.0.1
[google_api_ad_sense](#) 0.0.1
[google_api_ad_sense_host](#) 0.0.1
[google_api_admin](#) 0.0.1
[google_api_analytics_reporting](#) 0.0.1
[google_api_android_device_provisioning](#) 0.0.1
[google_api_android_enterprise](#) 0.0.1
[google_api_android_management](#) 0.0.1
[google_api_app_engine](#) 0.0.1
[google_api_app_state](#) 0.0.1
[google_api_apps_activity](#) 0.0.1
[google_api_big_query](#) 0.0.1
[google_api_big_query_data_transfer](#) 0.0.1
[google_api_blogger](#) 0.0.1
[google_api_books](#) 0.0.1
[google_api_calendar](#) 0.0.1
[google_api_civic_info](#) 0.0.1
[google_api_classroom](#) 0.0.1



Google Cloud Platform

google-cloud+hex@google.com
[GoogleCloudPlatform](#) on
[GitHub](#)

google_api_<name> https://github.com/GoogleCloudPlatform/elixir-google-api	Generated APIs
Tesla https://github.com/teamon/tesla	HTTP
Goth https://github.com/peburrows/goth	Authentication (service accounts)
OAuth2 https://github.com/scrogson/oauth2	Authentication (OAuth)
Poison https://github.com/devinus/poison	JSON encoding/decoding

Results

- Most APIs work
- Good enough developer experience
- Path to full support

Challenges

- Large initial investment
- Writing “generatable” code
- “Ugly” generated code
- Definition accuracy
- Testing

Challenges

- Large initial investment
- Writing “generatable” code
- **“Ugly” generated code**
- Definition accuracy
- Testing

Challenges

- Large initial investment
- Writing “generatable” code
- “Ugly” generated code
- **Definition accuracy**
- Testing

Challenges

- Large initial investment
- Writing “generatable” code
- “Ugly” generated code
- Definition accuracy
- **Testing**

Future Goals

- Idiomatic interfaces

Future Goals

- Idiomatic interfaces
- Common library

Future Goals

- Idiomatic interfaces
- Common library
- Publish OpenAPI v3

What can you do?

What can you do?

1. Define your APIs

What can you do?

1. Define your APIs
2. Generate your API clients

What can you do?

1. Define your APIs
2. Generate your API clients
3. Contribute back to existing tools

Thanks!

Jeff Ching

GitHub: [chingor13](#)

Code BEAM SF

March 16, 2018