Evolvable APIs with APIO Architect

Alejandro Hernandez @alejandrohdezma


Ask any doubt!

Vulcan?
APIO?
Hypermedia?

Liferay APIs in 2018

→ SOAP

→ JSONWS API (/api/jsonws)

→ JAX-RS (7.0)

...

APIO Goals

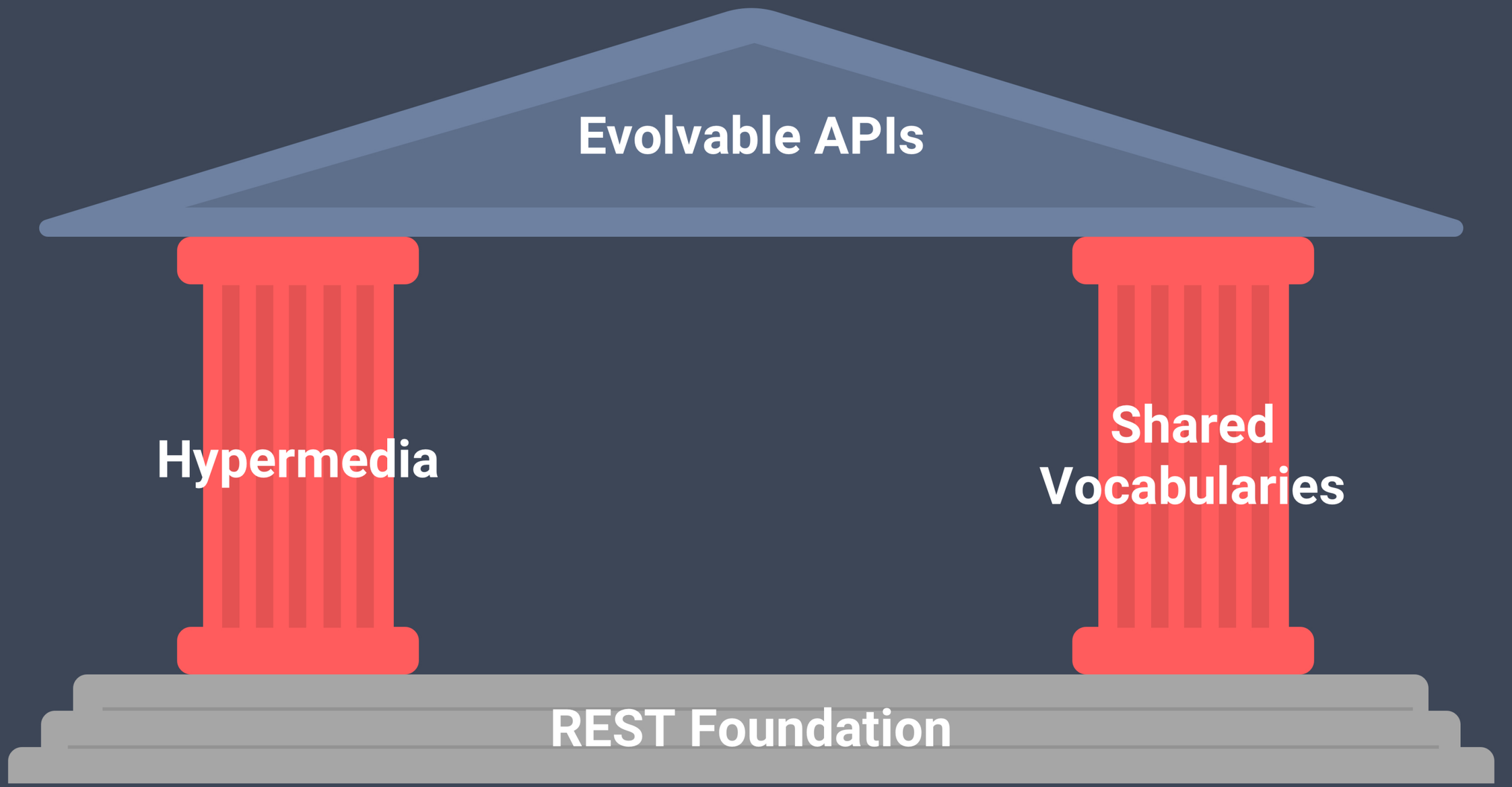
→ Easy versions

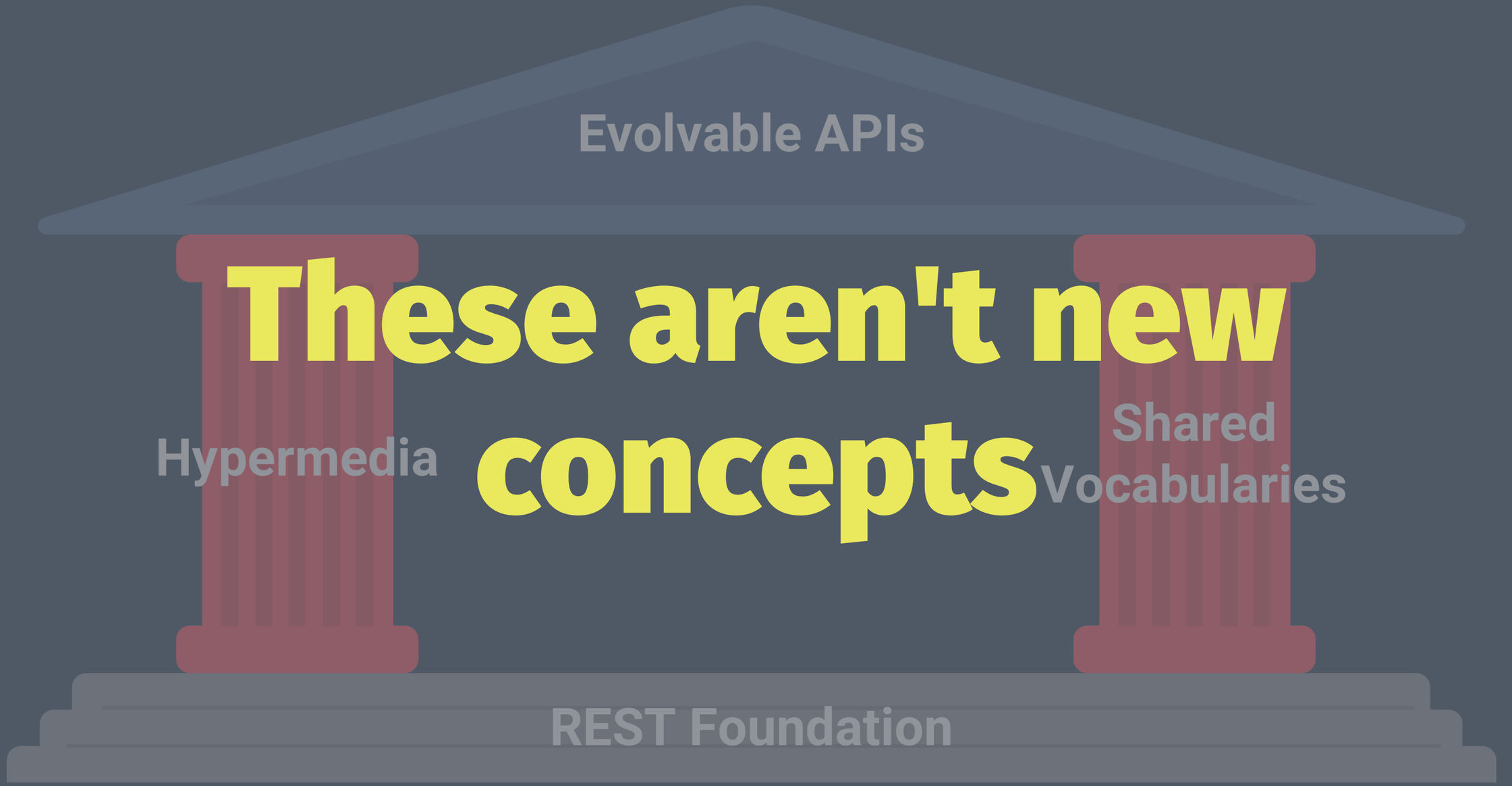
→ Code reuse

→ Auto-documentation

→ Fun!

How do we design APIs that are capable of evolving?


Evolvable APIs
Hypermedia
Shared
Vocabularies
REST Foundation



These aren't new concepts

REST style is an abstraction of the architectural elements within a distributed hypermedia system

— Roy Fielding, 2000

What's Hypermedia?

```
{
    "total": 43,
    "count": 30,
    "_links": {
        "collection": {
            "href": "http://apiosample.wedeploy.io/p/people"
        },
        "first": {
            "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"
        },
        "last": {
            "href": "http://apiosample.wedeploy.io/p/people?page=2&per_page=30"
        },
        "next": {
            "href": "http://apiosample.wedeploy.io/p/people?page=2&per_page=30"
        },
        "self": {
            "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"
        }
    },
}
```

Pagination (HAL)

```
{
    "total": 43,
    "count": 30,
    "_links": {
        "collection": {
            "href": "http://apiosample.wedeploy.io/p/people"
        },
        "first": {
            "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"
        },
        "last": {
            "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"
        },
        "self": {
            "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"
        }
    },
}
```

Standard link types (IANA Link Relations)

```
{
    "resources": {
        "blog-postings": {
            "href": "http://apiosample.wedeploy.io/p/blog-postings"
        },
        "people": {
            "href": "http://apiosample.wedeploy.io/p/people"
        }
    }
}
```

Entrypoint with all the "root" APIs (JSON HOME)

```
{
    "@id": "http://apiosample.wedeploy.io/p/people",
    "@type": [
        "Collection"
    ],
    "members": [
        ...
    ],
    "numberOfItems": 10,
    "operation": [
        {
            "@id": "people/create",
            "@type": "Operation",
            "expects": "http://apiosample.wedeploy.io/f/c/people",
            "method": "POST"
        }
    ]
}
```

Actions (Hydra + JSON-LD)

```
{
    "@id": "http://apiosample.wedeploy.io/p/people",
    "@type": [
        "Collection"
    ],
    "members": [
        ...
    ],
    "numberOfItems": 10,
    "operation": [
        {
            "@id": "people/create",
            "@type": "Operation",
            "expects": "http://apiosample.wedeploy.io/f/c/people",
            "method": "POST"
        }
    ]
}
```

Actions (Hydra + JSON-LD)

```
{
    "@context": "http://www.w3.org/ns/hydra/context.jsonld",
    "@id": "http://apiosample.wedeploy.io/f/c/people",
    "@type": "Class",
    "description": "This form can be used to create or update a person",
    "supportedProperty": [
        {
            "@type": "SupportedProperty",
            "property": "#familyName",
            "readable": false,
            "required": true,
            "writeable": true
        },
        ...
    ],
    "title": "The person form"
}
```

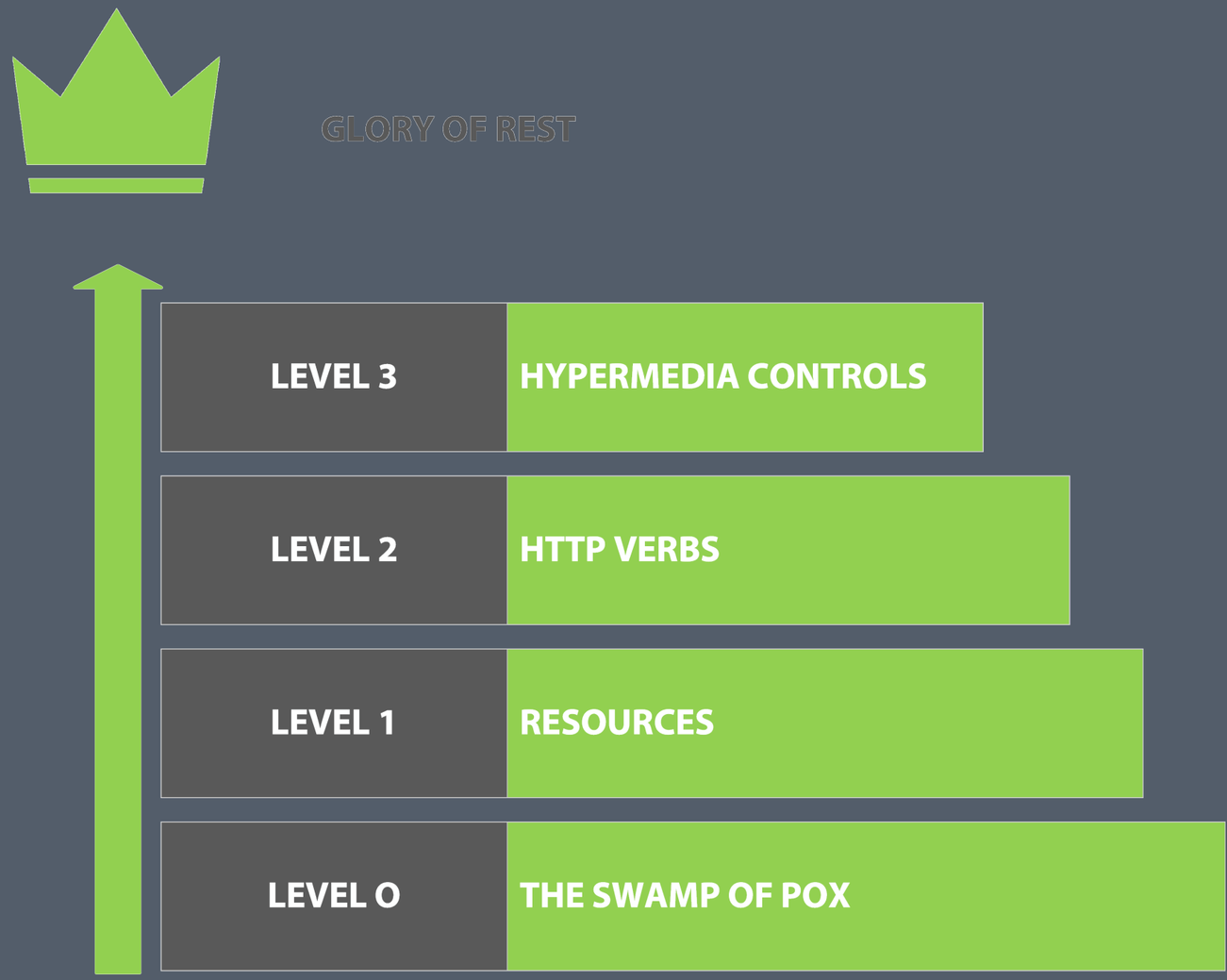
Forms (Hydra + JSON-LD)

```
{
    "headline": "A blog",
    "alternativeHeadline": "A subtitle",
    "_links": {
        "creator": {
            "href": "http://apiosample.wedeploy.io/p/people/1"
        },
        "self": {
            "href": "http://apiosample.wedeploy.io/p/blog-postings/12"
        }
    },
}
```

Links to other resources (HAL)

REST "well" done (Richardson Maturity Model)

— Martin Fowler's blog



Code time!

apio-architect-workshop

Problems creating APIs?

1 - Which format is better?

```
{
    "gender": "female",
    "familyName": "Hart",
    "givenName": "Sophia",
    "jobTitle": "Junior Executive",
    "name": "Sophia Hart",
    "birthDate": "1965-04-12T00:00Z",
    "email": "sophia.hart@example.com",
    "_links": {
        "self": {
            "href": "http://localhost:8080/o/api/p/people/30723"
        }
    }
}
```


HAL

```
{
    "class": "BlogPosting",
    "properties": {
        "headline": "Hello DEVCON!",
        "content": "The content of this blog posting"
    },
    "actions": [
        {
            "name": "delete-blog-posting",
            "title": "Delete Blog Posting",
            "method": "DELETE",
            "href": "http://localhost:8080/o/p/blogs/32400"
        }
    ],
}
```

SIREN

```
{
    "gender": "female",
    "familyName": "Hart",
    "givenName": "Sophia",
    "jobTitle": "Junior Executive",
    "name": "Sophia Hart",
    "birthDate": "1965-04-12T00:00Z",
    "email": "sophia.hart@example.com",
    "@id": "http://localhost:8080/o/api/p/people/30723",
    "@type": "Person",
    "@context": "http://schema.org"
}
```


JSON-LD

All of them

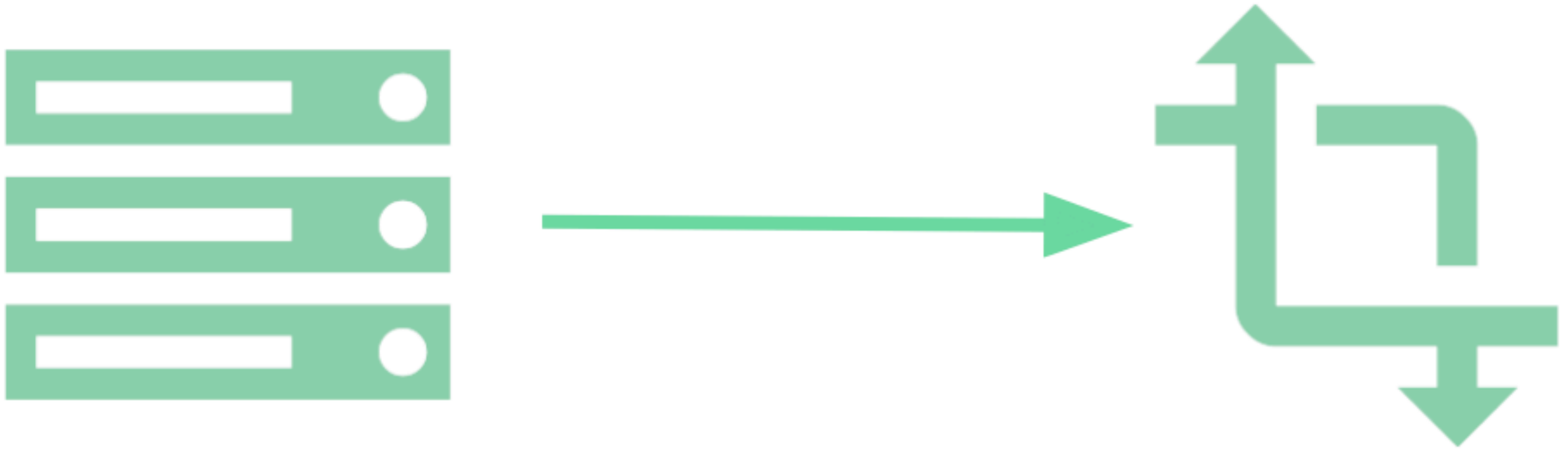
All of them

or none

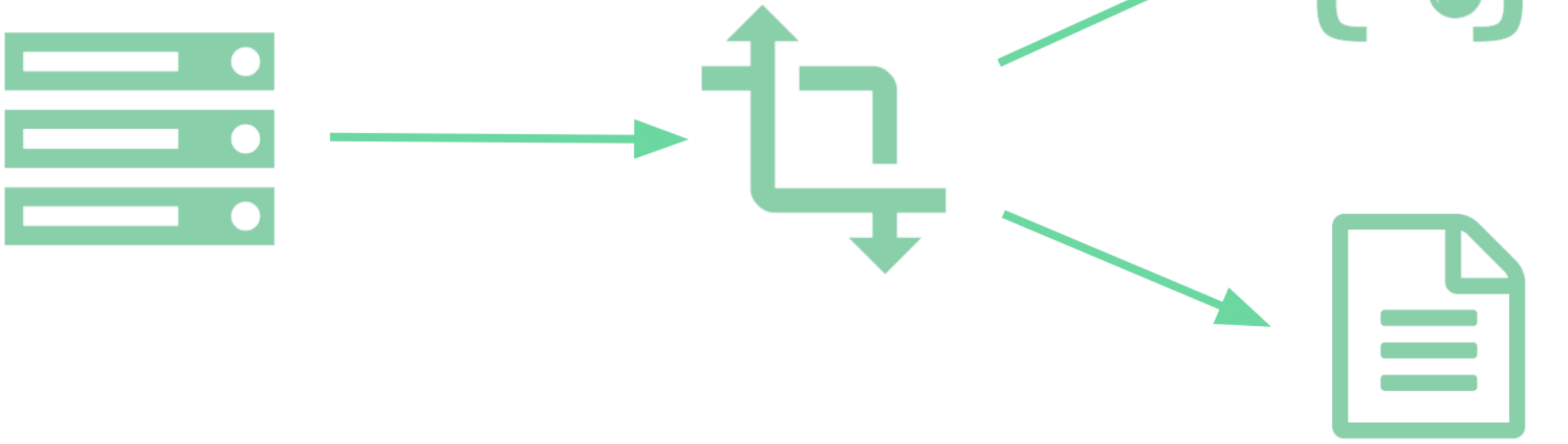
It depends on your case

Whats the solution then?

APIO Architect



Write to one format



```java
@Component(immediate = true)
public class BlogPostingResource
    implements CollectionResource<BlogsEntry, Long, ??> {

    @Override
    public String getName() {
        return "blog-postings";
    }

    @Override
    public Representor<BlogsEntry, Long> representor(
        Builder<BlogsEntry, Long> builder) {

        return builder.types(
            "BlogPosting"
        ).identifier(
            BlogsEntry::getEntryId
        ).build();
    }

}
```

```java
@Component(immediate = true)
public class BlogPostingResource
    implements CollectionResource<BlogsEntry, Long, ??> {

    @Override
    public String getName() {
        return "blog-postings";
    }

    @Override
    public Representor<BlogsEntry, Long> representor(
        Builder<BlogsEntry, Long> builder) {

        return builder.types(
            "BlogPosting"
        ).identifier(
            BlogsEntry::getEntryId
        ).build();
    }

}
```

```
public interface BlogPostingId extends Identifier<Long> {
}
```

```java
@Component(immediate = true)
public class BlogPostingResource
    implements CollectionResource<BlogsEntry, Long, BlogPostingId> {

    @Override
    public String getName() {
        return "blog-postings";
    }

    @Override
    public Representor<BlogsEntry, Long> representor(
        Builder<BlogsEntry, Long> builder) {

        return builder.types(
            "BlogPosting"
        ).identifier(
            BlogsEntry::getEntryId
        ).build();
    }

}
```

```
public Representor<BlogsEntry, Long> representor(
    Builder<BlogsEntry, Long> builder) {

    return builder.types(
        "BlogPosting"
    ).identifier(
        blogsEntry -> blogsEntry.getEntryId()
    ).build();
}
```

The Representor

```
public Representor<BlogsEntry, Long> representor(
    Builder<BlogsEntry, Long> builder) {

    return builder.types(
        "BlogPosting"
    ).identifier(
        blogsEntry -> blogsEntry.getEntryId()
    ).addDate(
        "displayDate", BlogsEntry::getDisplayDate
    ).addLink(
        "license", "https://creativecommons.org/licenses/by/4.0"
    ).addString(
        "alternativeHeadline", BlogsEntry::getSubtitle
    ).addString(
        "articleBody", BlogsEntry::getContent
    ).addString(
        "description", BlogsEntry::getDescription
    ).addString(
        "headline", BlogsEntry::getTitle
    ).build();
}
```

The Representor

Mappers:

→ JSON

→ HAL

→ JSON+LD

→ HTML?

→ ...

Problems creating APIs?

2 - Routes

What do we want to achieve?

Decouple route and representation code

Reuse code to represent CRUD

Clearly express the supported routes

CollectionRoutes.Builder

```
@Override
public CollectionRoutes<BlogsEntry> collectionRoutes(
    CollectionRoutes.Builder<BlogsEntry> builder) {

    return builder.addGetter(
        this::_getPageItems
    ).build();
}

private PageItems<BlogsEntry> _getPageItems(Pagination pagination) {
    List<BlogsEntry> blogsEntries = _blogsService.getGroupEntries(
        20143, 0, pagination.getStartPosition(),
        pagination.getEndPosition());

    int count = _blogsService.getGroupEntriesCount(20143, 0);

    return new PageItems<>(blogsEntries, count);
}

@Reference
private BlogsEntryService _blogsService;
```

```
@Override
public CollectionRoutes<BlogsEntry> collectionRoutes(
    Builder<BlogsEntry, Long> builder) {

    return builder.addGetter(
        this::_getPageItems
    ).addCreator(
        this::_addBlogsEntry, BlogPostingForm::buildForm
    ).build();
}
```

```
{
    "@context": "http://www.w3.org/ns/hydra/context.jsonld",
    "@id": "http://apiosample.wedeploy.io/f/c/blog-postings",
    "@type": "Class",
    "description": "This form can be used to create or update a blog posting",
    "supportedProperty": [
        {
            "@type": "SupportedProperty",
            "property": "#headline",
            "readable": false,
            "required": true,
            "writeable": true
        },
        ...
    ],
    "title": "The blog posting form"
}
```

What about item routes?

ItemRoutes.Builder

```
@Override
public ItemRoutes<BlogsEntry> itemRoutes(
    Builder<BlogsEntry, Long> builder) {

    return builder.addGetter(
        this::_getBlogsEntry
    ).addRemover(
        this::_deleteBlogsEntry
    ).addUpdater(
        this::_updateBlogsEntry, BlogsEntryForm::buildForm
    ).build();
}
```

Problems creating APIs?

3 - Converting fields, links, dates

Why?

Types! (dates, strings, numbers)

Links

Shared vocabularies!

BlogPosting

Canonical URL: http://schema.org/BlogPosting

Thing > CreativeWork > Article > SocialMediaPosting > BlogPosting

A blog post.

Usage: Over 1,000,000 domains

[more...]

Property	Expected Type	Description
Properties from SocialMediaPosting		
sharedContent	CreativeWork	A CreativeWork such as an image, video, or audio clip shared as part of this posting.
Properties from Article		
articleBody	Text	The actual body of the article.
articleSection	Text	Articles may belong to one or more 'sections' in a magazine or newspaper, such as Sports, Lifestyle, etc.
pageEnd	Integer or Text	The page on which the work ends; for example "138" or "xvi".
pageStart	Integer or Text	The page on which the work starts; for example "135" or "xiii".
pagination	Text	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55" or "10-12, 46-49".
	SpeakableSpecification or URL	Indicates sections of a Web page that are particularly 'speakable' in the sense of being highlighted as being especially appropriate for text-to-speech conversion. Other sections of a page may also be usefully spoken in particular circumstances; the 'speakable' property serves to indicate the parts most likely to be generally useful for speech. The *speakable* property can be repeated an arbitrary number of times, with three kinds of possible 'content-locator' values:

BlogsEntry	BlogPosting
getTitle()	headline
getContent()	articleBody
getSubtitle()	alternativeHeadline
getLastPublishedDate()	lastPublishedDate

Mapping BlogsEntry to BlogPosting

```
public Representor<BlogsEntry, Long> representor(
    Builder<BlogsEntry, Long> builder) {

    return builder.types(
        "BlogPosting"
    ).identifier(
        blogsEntry -> blogsEntry.getEntryId()
    ).addDate(
        "displayDate", BlogsEntry::getDisplayDate
    ).addLink(
        "license", "https://creativecommons.org/licenses/by/4.0"
    ).addString(
        "alternativeHeadline", BlogsEntry::getSubtitle
    ).addString(
        "articleBody", BlogsEntry::getContent
    ).addString(
        "description", BlogsEntry::getDescription
    ).addString(
        "headline", BlogsEntry::getTitle
    ).build();
}
```

Problems creating APIs?

4 - Relations between resources

Embedded resources (save requests)

Navigation between resources

```
public class PersonResource
    implements CollectionResource<User, Long, PersonId> {

    @Override
    public String getName() {
        return "people";
    }

    @Override
    public Representor<User, Long> representor(
        Representor.Builder<User, Long> representorBuilder) {

        return representorBuilder.types(
            "Person"
        ).identifier(
            User::getUserId
        ).addString(
            "additionalName", User::getMiddleName
        ).addString(
            "email", User::getEmailAddress
        ).addString(
            "familyName", User::getLastName
        ).addString(
            "givenName", User::getFirstName
        ).addString(
            "jobTitle", User::getJobTitle
        ).build();
    }

}
```

```
public Representor<BlogsEntry, Long> representor(
    Builder<BlogsEntry, Long> builder) {

    return builder.types(
        "BlogPosting"
    ).identifier(
        BlogsEntry::getEntryId
    ).addDate(
        "displayDate", BlogsEntry::getDisplayDate
    ).addLinkedModel(
        "creator", PersonId.class, blogsEntry -> blogsEntry.getUserId()
    ).addString(
        "articleBody", BlogsEntry::getContent
    ).addString(
        "headline", BlogsEntry::getTitle
    ).build();
}
```

What else?

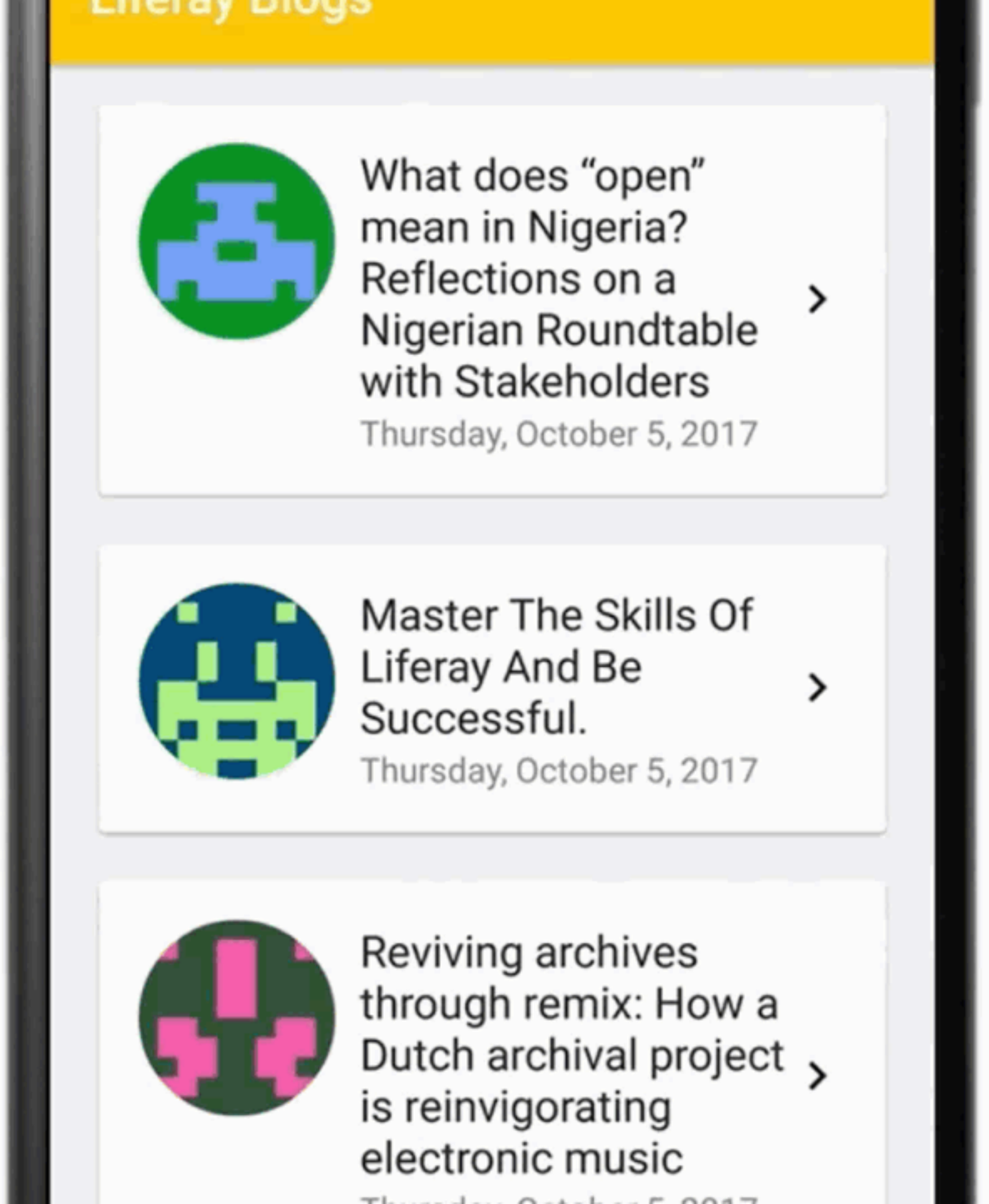
I want more

API Consumer

```
val id =  "http://.../o/api/p/web-sites/5745/blogs"

thingScreenlet.load(id, credentials)
```

Clients for Java (+Android), iOS and Javascript (coming...)




#apio-users

Questions?

Evolvable APIs with APIO Architect

Alejandro Hernández @alejandrohdezma