RESTful or RESTless

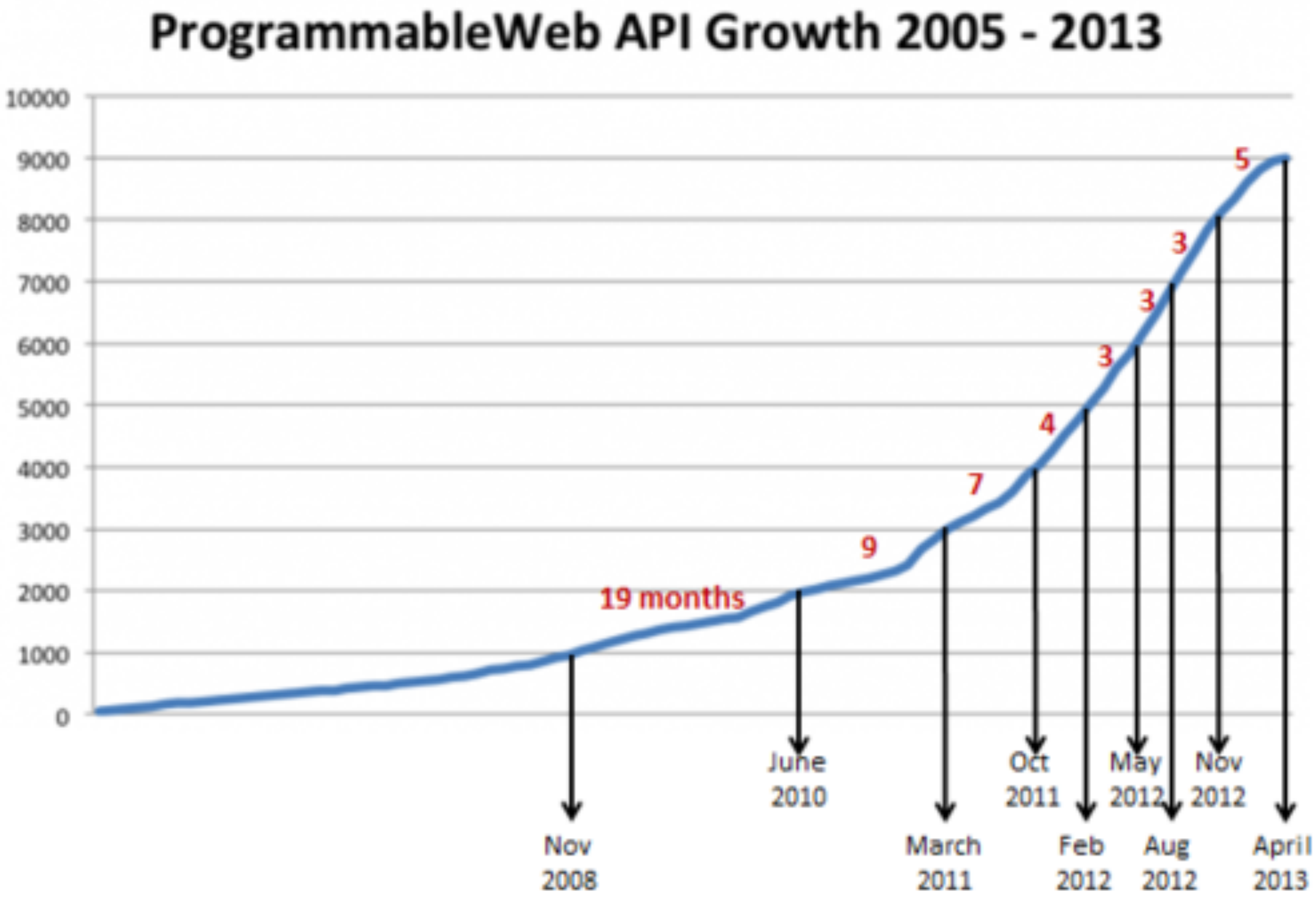
Current State of Today's Top Web APIs

Frederik Buelthoff, **Maria Maleshkova**

AIFB, Karlsruhe Institute of Technology (KIT), Germany

[1]

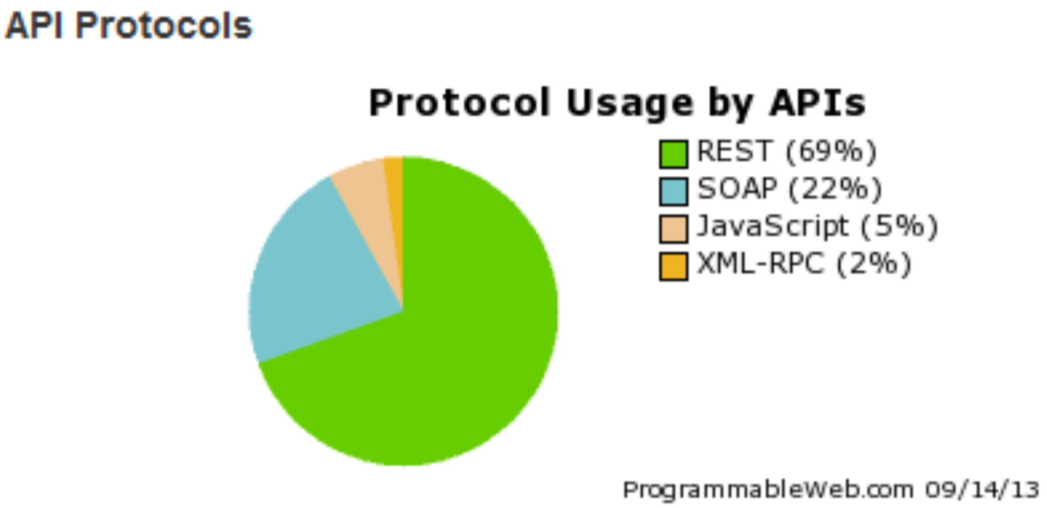
Growing Number of Web APIs



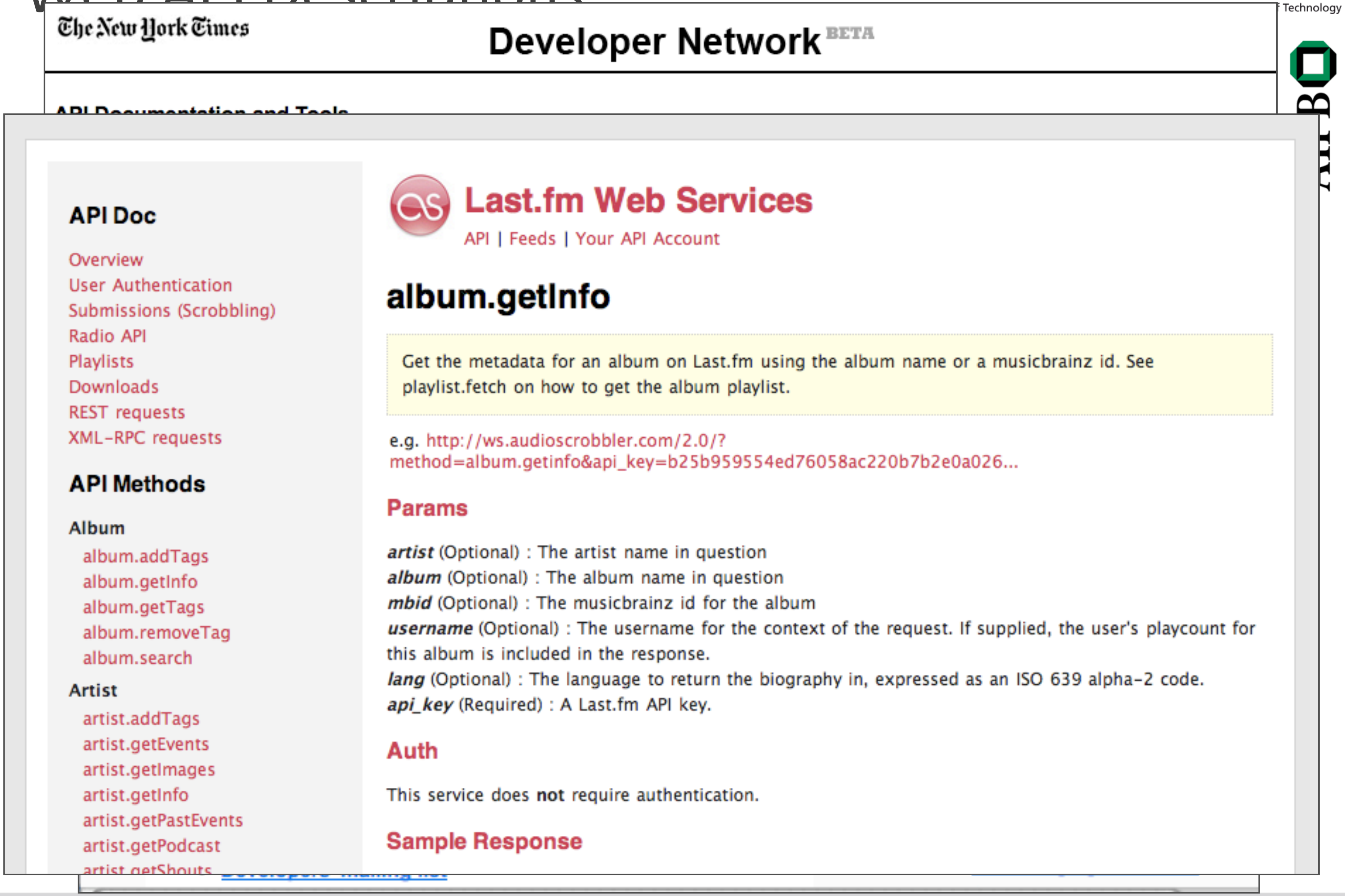
Challenges

- Scalability
- Automation
- Managing complexity

???



Web API Descriptions



The New York Times Developer Network BETA

API Documentation and Tools

Last.fm Web Services

API | Feeds | Your API Account

API Doc

- Overview
- User Authentication
- Submissions (Scrobbling)
- Radio API
- Playlists
- Downloads
- REST requests
- XML-RPC requests

API Methods

Album

- album.addTags
- album.getInfo
- album.getTags
- album.removeTag
- album.search

Artist

- artist.addTags
- artist.getEvents
- artist.getImages
- artist.getInfo
- artist.getPastEvents
- artist.getPodcast
- artist.getShouts

album.getInfo

Get the metadata for an album on Last.fm using the album name or a musicbrainz id. See playlist.fetch on how to get the album playlist.

e.g. http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=b25b959554ed76058ac220b7b2e0a026...

Params

artist (Optional) : The artist name in question
album (Optional) : The album name in question
mbid (Optional) : The musicbrainz id for the album
username (Optional) : The username for the context of the request. If supplied, the user's playcount for this album is included in the response.
lang (Optional) : The language to return the biography in, expressed as an ISO 639 alpha-2 code.
api_key (Required) : A Last.fm API key.

Auth

This service does **not** require authentication.

Sample Response

Developers' mailing list

What is a Web API?

- ???
- No established definition
- REST principles as foundation

→ Web API Survey

Survey Setup

- http://www.programmableweb.com
- Conducted during December 2013/January 2014
- Single domain expert
- 45 Web APIs
  - Use number of mashups as a metric for the popularity of the Web API (biased towards older APIs)
  - Alexa.com rank and number of tagged questions on StackOverflow.com

Amazon Product Advertising, Amazon S3, BitBucket, Azure (Blob Service), Balanced Payments, Bing Maps REST Services, Bitly, Box, del.icio.us, Disqus, DocuSign Enterprise, Dropbox (Core API), eBay (Shopping API), Etsy, Eventful, Facebook (Graph API), Flickr, Foursquare, Freebase (Search/Reconcile), Geonames, GitHub, Google Custom Search, Google Maps APIWeb Services, Google Places API, Groupon, Heroku, Instagram, Last.fm, LinkedIn, OpenStreetMap (Editing API), Panoramio, Paypal, Reddit, Salesforce, Tropo, Tumblr, Twilio, Twitpic, Twitter, Wikipedia/Mediawiki, Yahoo! BOSS, Yahoo! BOSS Geo, Yammer, Yelp, Youtube

Analyzed Characteristics

- **General Web API Information** – the APIs size in terms of operations, availability of alternative protocols, interface descriptions
- **URL and Resource Links** – URL design, use of links between API resources
- **HTTP Use** – HTTP methods, support for alternative HTTP methods, use of HTTP error status codes, caching mechanisms
- **Input and Output Data** – way of transmitting input data, types of input, output formats
- **Security and Policies** – types of authentication, limitations on usage
- **Common Design Decision** – how are versioning and the selection of the output format realized

General Web API Information

- Measuring size/complexity
  - The majority (62%) of the APIs had between eleven and one hundred operations (38% in the 11-50 and 24% in the 51-100 range)
  - Remaining Web APIs equally divided into smaller (less than 11 operations, 20%) and larger (>100 operations, 18%)
  - Only two entries (4%) provided a single operation
- Alternative protocols
  - Only a small percentage (20%) of the Web APIs provided the same service using alternative protocols
  - Flickr is available through SOAP, XML-RPC and REST
- Interface descriptions in a machine-readable format
  - Web APIs (11%) – using a custom format, JSON Hyper-Schema, WSDL
- Links to related resources embedded in the response data of Web APIs (an alternative to interface descriptions) available in eight cases (18%)

General Web API Information - Summary

- Once an HTTP-based Web API is available, providers tend to move away from previous interaction protocol implementations (e.g. SOAP)

- Machine-interpretable interface description formats are rather an exception than a rule
  - Most providers still prefer to document APIs directly as part of webpages

URLs and Resource Links

- HATEOAS in the URL design
  - Design or structure of the URLs remains a good indicator for the type of the Web API.
  - The availability of resource links was previously presented as part of the analysis on interface

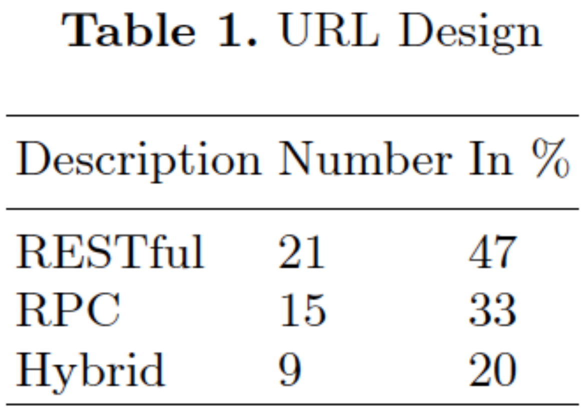


Table 1. URL Design

Description	Number	In %
RESTful	21	47
RPC	15	33
Hybrid	9	20

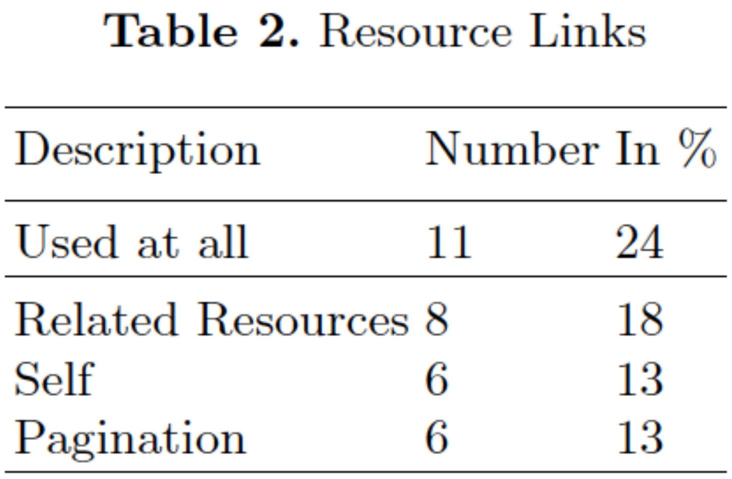


Table 2. Resource Links

Description	Number	In %
Used at all	11	24
Related Resources	8	18
Self	6	13
Pagination	6	13

- Web APIs with self links include the URL of resources as part of their representation
- Pagination links provide the client with precomposed URLs for paging

URLs and Resource Links – Summary

- HATEOAS remains one of the most poorly supported constraints of the REST architecture
  - Less than a fifth of the analyzed Web APIs providing links to related resources
  - Notable exceptions include PayPal and Github

HTTP Use

- Two most commonly used HTTP verbs are

  GET and POST
- The most popular way for indicating the HTTP verb is via a query parameter

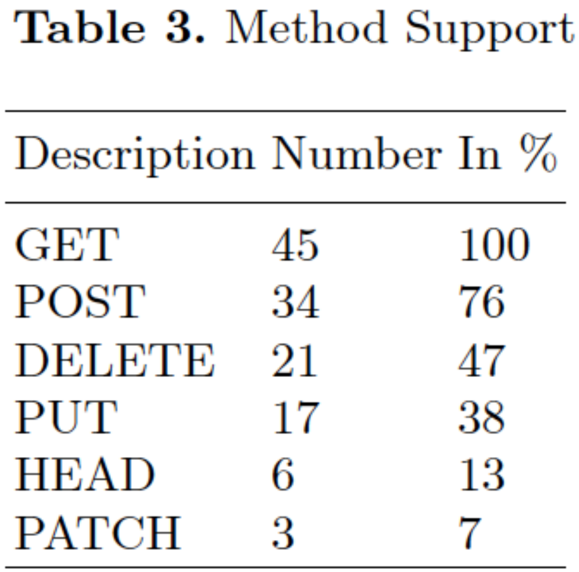


Table 3. Method Support

Description	Number	In %
GET	45	100
POST	34	76
DELETE	21	47
PUT	17	38
HEAD	6	13
PATCH	3	7

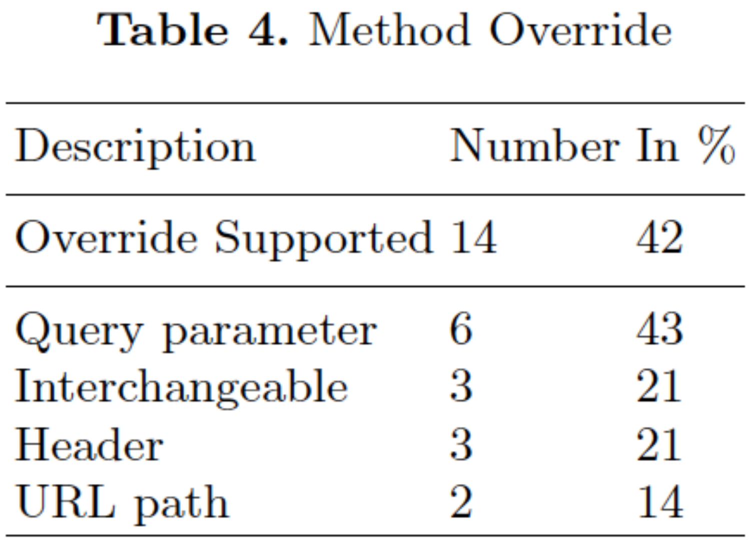


Table 4. Method Override

Description	Number	In %
Override Supported	14	42
Query parameter	6	43
Interchangeable	3	21
Header	3	21
URL path	2	14

HTTP Use

- 71% use the HTTP status codes to indicate an error

- 27% of the Web APIs explicitly state support for caching
  - Test invocations show additional six Web APIs with caching support, without having documented it

Input and Output Data

- Ways for sending the input
- Type of input

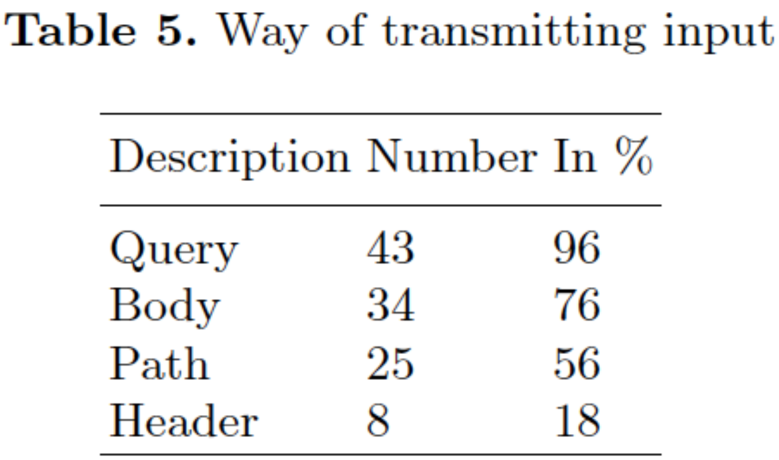


Table 5. Way of transmitting input

Description	Number	In %
Query	43	96
Body	34	76
Path	25	56
Header	8	18

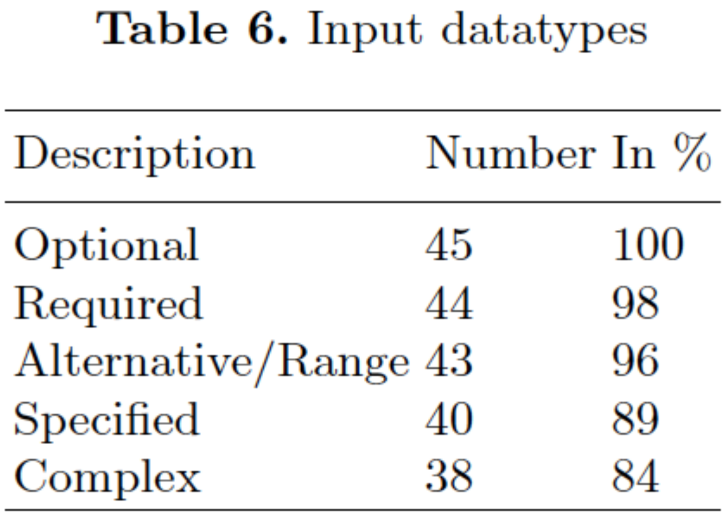


Table 6. Input datatypes

Description	Number	In %
Optional	45	100
Required	44	98
Alternative/Range	43	96
Specified	40	89
Complex	38	84

- As output format Web APIs most commonly (89%) support JSON
- XML remains the second most used data format (58%)

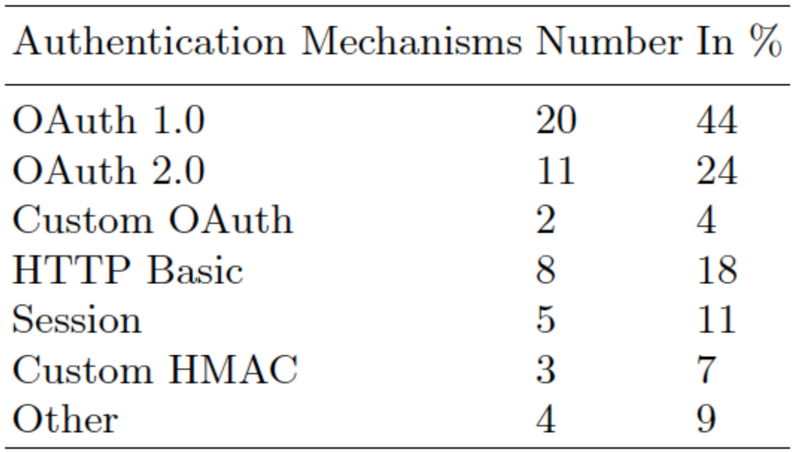
Input and Output Data – Summary

- Preparing the input in the right format requires additional effort
- There is no general consensus on how to format frequently occurring input (such as date and time)
  - Manual effort (for service invocation and composition)
- JSON and XML the two main established output data formats, with JSON rapidly gaining on importance

Security and Policies

- The most common way of identifying the client application/user is via an **API key**
- Most Web APIs (89%) state and implement rate limitations

Table 7. Common Web API Authentication Approaches



Authentication Mechanisms	Number	In %
OAuth 1.0	20	44
OAuth 2.0	11	24
Custom OAuth	2	4
HTTP Basic	8	18
Session	5	11
Custom HMAC	3	7
Other	4	9

- The majority of Web APIs use authentication
  - Requiring developers to 1) register their application in advance 2) tackle individual authentication mechanisms
- OAuth has the potential to emerge as universally adopted standard for authentication

Common Design Decisions

- Versioning
  - This was addressed by 73% of the examined Web APIs

- Way of selecting the output format

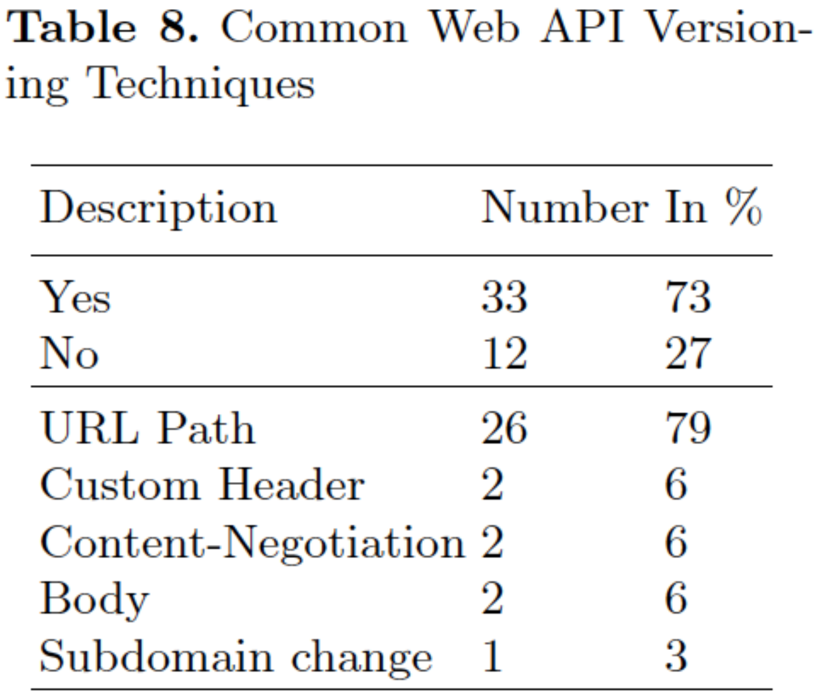


Table 8. Common Web API Versioning Techniques

Description	Number	In %
Yes	33	73
No	12	27
URL Path	26	79
Custom Header	2	6
Content-Negotiation	2	6
Body	2	6
Subdomain change	1	3

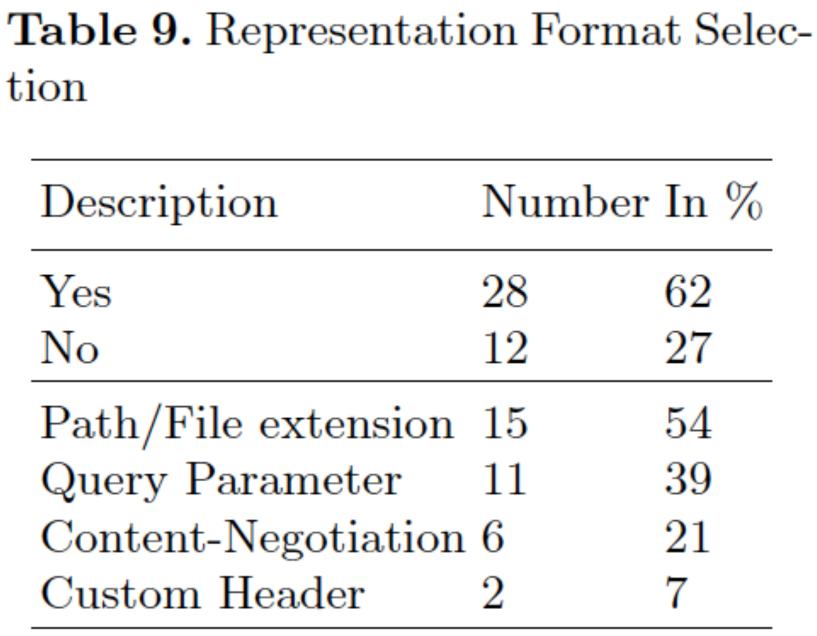


Table 9. Representation Format Selection

Description	Number	In %
Yes	28	62
No	12	27
Path/File extension	15	54
Query Parameter	11	39
Content-Negotiation	6	21
Custom Header	2	7

RESTless!

- Web APIs feature a large amount of heterogeneity
  - Require more manual effort to smooth over differences in implementations
- Some more understandable concepts (such as using the HTTP verbs) have gained widespread adoption
- Other concepts (such as resource linking (HATEOAS)) are hardly ever applied

- [1] RESTful Java Web Services with NetBeans, Jersey and Tomcat, http://vichargrave.com/restful-web-service-development-with-netbeans-and-tomcat-tutorial/