

Opportunities in Software Engineering Research for Web API Consumption

WAPI'17

Erik Wittern, Annie Ying, Yunhui Zheng, Jim A. Laredo, Julian Dolby, Christopher C. Young, Aleksander A. Slominski

IBM T.J. Watson Research Center

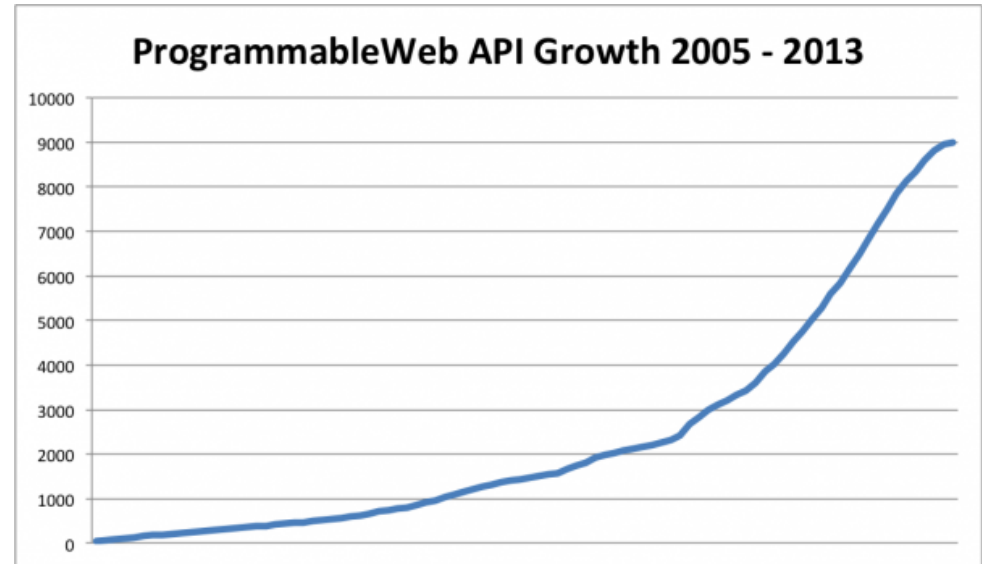


Part 1: Web APIs and challenges for consumption



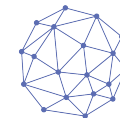
Web APIs are ubiquitous

- The number of web APIs is continuously growing
- Enable programmatic interaction with remote resources
- ...using of existing, ubiquitous Web technologies
(mostly HTTP + JSON / XML)
- APIs are of great importance for industry to create application ecosystems
- Focus on simplicity & flexibility – few conventions or rules
(contrast with SOAP/WS-*)



Source: ProgrammableWeb.com, now > 17k

apigee



IBM API Connect

StrongLoop



mashape

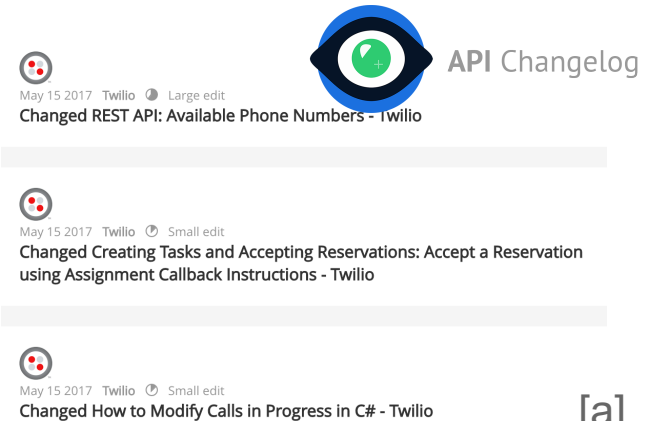


Consumption is challenging, though

String-based interface;
no type-checking

```
request({  
  url: 'http://api.example.com/v1/events',  
  method: 'post',  
  body: JSON.stringify({name: 'WAPI \'17'})  
}, (err, response, strData) => {  
  let responseData = JSON.parse(strData)  
  // ...  
})
```

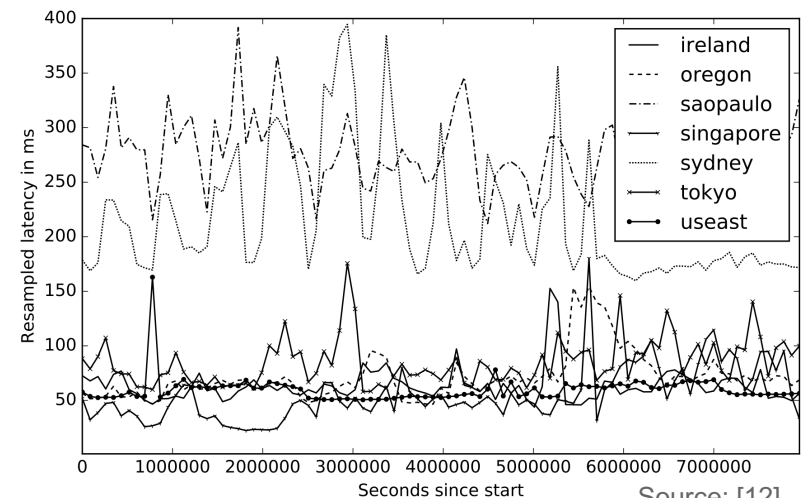
Frequent changes



APIs are controlled by third-party



Remote calls with varying QoS



Opinion: Gap between practice & challenges and software engineering research

stripe



Challenges



SE research

Part 2: IDE support for checking web API requests

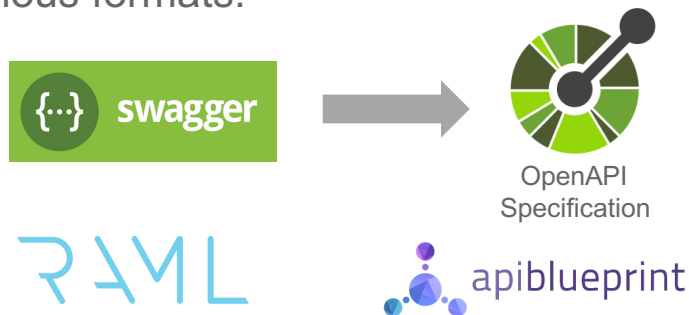


Specifications are central to support web API consumption

- Specifications describe possible interactions with an API – they depict the *contract* between client and API

- Machine-readable

- Various formats:



- They describe...
 - Base URL
 - Endpoints (path template + HTTP method)
 - Data required by / returned from endpoints
 - Additional parameters (e.g. in query)
 - Other constraints like headers, authentication, rate limits...

```
swagger: '2.0'
host: api.twilio.com
basePath: /2010-04-01
schemes:
  - http
paths:
  /Accounts/{userId}/Messages.json:
    post:
      description: 'Send a message'
      parameters:
        - in: path
          name: userId
          required: true
          type: string
        - in: body
          name: body
          schema:
            $ref: '#/definitions/Message'
        - in: query
          name: 'retry'
          required: false
          type: boolean
      ...
definitions:
  Message:
    type: object
    properties:
      from: ...
      to: ...
      body: ...
    required: from, to, body
```

Excerpt of an OpenAPI Specification

Overview of the approach

```
1 var getPicturesForTag = function (tag) {
2   var query = {
3     count: 10
4   }
5   var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'
6   var settings = {
7     method: 'GET',
8     url: url,
9     data: query
10  }
11  sendRequest(settings)
12 }
13
14 var sendRequest = function (settings) {
15   $.ajax(settings).done(function (response) {
16     console.log(response)
17   })
18 }
```

1. Extract request

**Details: ICSE
Web applications
Thursday 11:00-12:30**

→ = data flow
····· = control flow

**2. Match request
against
specifications**

3. Report results

instagram_example.js — ~/Development/apiharmony/apih-atom-advice

API Harmony detected 1 errors

- Method "post" not found for path "/tags/{tag-name}/media/recent".

```
1 var getPicturesForTag = function (tag) {
2   var query = {
3     count: 10
4   }
5   var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'
6   var settings = {
7     method: 'POST',
8     url: url,
9     data: query
10  }
11  sendRequest(settings)
12 }
13
14 var sendRequest = function (settings) {
15   $.ajax(settings).done(function (response) {
16     console.log(response)
17   })
18 }
```

File 0 Project 0 ✓ No Isss LF ⚠ 1 deprecation UTF-8 JavaScript master +21 3 u

(a)

instagram_example.js — ~/Development/apiharmony/apih-atom-advice

API Harmony approves of this request

Matching OpenAPI Specification "Instagram" found.

- Request matches path "/tags/{tag-name}/media/recent".
- Method "get" available for path "/tags/{tag-name}/media/recent".
- All required query parameters are present.
- Payload is in the right format.

```
1 var getPicturesForTag = function (tag) {
2   var query = {
3     count: 10
4   }
5   var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'
6   var settings = {
7     method: 'GET',
8     url: url,
9     data: query
10  }
11  sendRequest(settings)
12 }
13
14 var sendRequest = function (settings) {
15   $.ajax(settings).done(function (response) {
16     console.log(response)
17   })
18 }
```

File 0 Project 0 ✓ No Isss LF ⚠ 1 deprecation UTF-8 JavaScript master +21 3 u

(b)

Example: Request to Medical Lab Inferencing Service

Video: <https://youtu.be/8lJKs7rMjJI>



Many research opportunities

- **Generation / maintenance of specifications**
 - From dynamic traces [7] or via proxies [21]
 - From documentation
 - Through code annotations [b]
- **Static checking of code** [8] [*this work*]
 - During development
 - On existing code bases
- **Automatic testing of APIs**
- **Mining API usage**
 - How are APIs used in the wild?
 - Lessons learned
- **QoS measurement & mitigation** [12]
- **Emerging API paradigms**
 - GraphQL / Apollo (Facebook, GitHub...)
 - Falcor (Netflix...)
- ...





@apiHarmony



<http://ibm.biz/apiharmony>
<http://www.apiful.io>



witternj@us.ibm.com
annie.ying@gmail.com
zhengyu@us.ibm.com
laredoj@us.ibm.com
dolby@us.ibm.com
aslom@us.ibm.com

Discussion points

- Why has web API consumption barely been subject of SE research?
- And how can we change this situation?
- Which existing SE tools & methods from library APIs can be used in the context of web APIs?



Detect inconsistencies between analysis results and specifications

```
{
  "request": {
    "data": {},
    "success": "JSFunction",
    "error": "JSFunction",
    "type": "GET",
    "url": "https://api.instagram.com/v1
           /tags/<$global#tag$>
           /media/recent?count=10"
  }
}
```

Example data produced by analysis

Procedure

1. Find all specifications that match **base URL**
2. Check usage of **schema**
3. Select specifications that also match **path**
4. Select specifications that also match **method**
5. Determine whether URL contains **required query parameters***
6. Determine whether payload matches the required **schema***

```
swagger: '2.0'
info: ...
host: api.instagram.com
basePath: /v1
schemes:
  - https
paths:
  /tags/{tag_id}/media/recent:
    get:
      description: 'Get recent media for tag'
      parameters:
        - in: query
          name: count
          required: true
          type: number
      responses:
        '200':
          schema:
            $ref: '#/definitions/Message'
definitions:
  Message:
    type: object
    properties:
      from: ...
      to: ...
      body: ...
    required: from, to, body
```

Example specification

System context: API Harmony - *find, learn about, and use web APIs*

The screenshot displays the API Harmony web application. At the top, there is a search bar with the placeholder text "Search for APIs...". Below the search bar, a "Welcome to API Harmony!" message is visible. The main content area shows search results for the "Instagram" API. On the left, a sidebar lists various API categories with their respective usage counts: GEOGRAPHIES (2), LOCATIONS (73), MEDIA (167), COMMENTS (3), LIKES (3), TAGS (380), and USERS (410). The main panel displays the "Instagram" API details, including its base URL, a description, and a "DOWNLOAD OPENAPI SPECIFICATION..." button. A "Request" section shows a JavaScript + jQuery code snippet for making an API call. A "Responses" section shows a JSON response example. A "Parameters" section lists query parameters like "lat" and "lng" with their descriptions and usage counts. A "Top response fields" section lists fields like "data", "responseText", and "data.length" with their usage counts. At the bottom, a box indicates the "Encryption scheme: HTTPS" and the "API: Instagram".

Search for APIs...

Welcome to API Harmony!

Find APIs: API Harmony currently lists 1,234 APIs from [APIs.guru](#), from crawling the web, and from other sources.

Learn about APIs: API Harmony provides information about APIs, including their documentation, and how others use APIs in their applications.

Use APIs: API Harmony contains information about how to use APIs, including GitHub projects using it, and other resources.

Instagram

API Information

- Characteristics & Relations
- StackOverflow questions
- Usages on GitHub
- Client Libraries on NPM

Endpoints

- GEOGRAPHIES (2) usages
- LOCATIONS (73) usages
- MEDIA (167) usages
- COMMENTS (3) usages
- LIKES (3) usages
- TAGS (380) usages
- USERS (410) usages

Base URL: <https://api.instagram.com/v1>

Instagram is beautiful...

[Contact URL](#) | [External Swagger URL](#) | [Documentation](#)

[DOWNLOAD OPENAPI SPECIFICATION...](#)

Characteristics & Relations

Request

JavaScript + jQuery

```
var settings = {
  "async": true,
  "crossDomain": true,
  "url": "https://api.instagram.com/v1/media/search?lat=40.7128&lng=-87.9512",
  "method": "GET",
  "headers": {
    "access_token": "REPLACE_KEY_VALUE"
  }
}

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

Responses

200

Found media resources (without likes information) in a given area.

Example Schema

```
{
  "data": [
    {}
  ],
  "meta": {
    "code": 0
  }
}
```

Top response fields

- `data` (used 16 times)
- `responseText` (used 4 times)
- `data.length` (used 3 times)

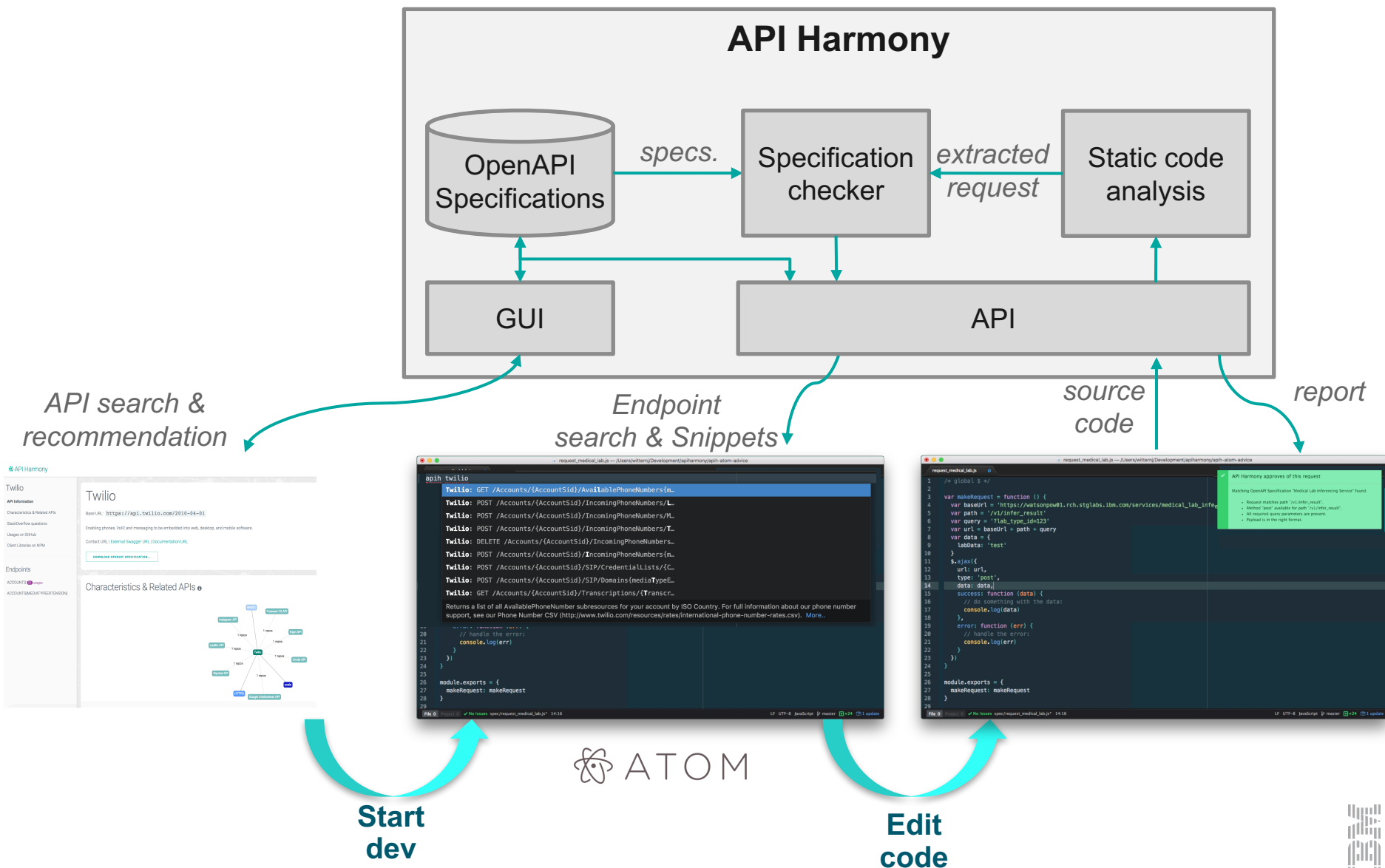
Parameters

QUERY

- `lat` - Latitude of the center search coordinate. If used, 'lng' is required. (used 44 times - set to `{latitude}` 4 times [1] [2] [3])
- `lng` - Longitude of the center search coordinate. If used, 'lat' is required. (used 44 times - set to `{longitude}` 4 times [1] [2] [3])

Encryption scheme: HTTPS | API: Instagram

System integration & developer work-flow



References (1/3)

- [a] Espinha, T., Zaidman, A., & Gross, H.-G. (2014). Web API growing pains: Stories from client developers and their code. (pp. 84–93). Presented at the CSMR-WCRE, IEEE.
- [b] <https://github.com/swagger-api/swagger-core>



References (2/3)

- [1] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, U. of California, Irvine, 2000.
- [2] IBM API Harmony. <https://apiharmony-open.mybluemix.net/>.
- [3] PublicAPIs. <https://www.publicapis.com/>.
- [4] ProgrammableWeb. <http://www.programmableweb.com/>.
- [5] M. Slee, A. Agarwal, and M. Kwiatkowski, “Thrift: Scalable cross-language services implementation,” *Facebook White Paper*, 2007.
- [6] M.P.Robillard,E.Bodden,D.Kawrykow,M.Mezini,andT.Ratchford, “Automated API property inference techniques,” *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 613–637, 2013.
- [7] P. Suter and E. Wittern, “Inferring Web API Descriptions From Usage Data,” in *Proc. of the Workshop on Hot Topics in Web Systems and Technologies*, 2015.
- [8] E. Wittern, A. T. T. Ying, Y. Zheng, J. Dolby, and J. A. Laredo, “Statically Checking Web API Requests in JavaScript,” in *Proc. of ICSE, to appear*, 2017.
- [9] B. Dagenais and M. P. Robillard, “Recommending adaptive changes for framework evolution,” *ACM TOSEM*, vol. 20, no. 4, p. 19, 2011.
- [10] W. Wu, Y.-G. Gue´he´neuc, G. Antoniol, and M. Kim, “Aura: a hybrid approach to identify framework evolution,” in *Proc. of ICSE*, 2010, pp. 325–334.
- [11] Open API Initiative. <https://openapis.org/specification>.
- [12] D. Bermbach and E. Wittern, “Benchmarking web API quality,” in *Proc. of the International Conference in Web Engineering*, 2016, pp. 188–206.



References (3/3)

- [13] D. Li, S. Hao, J. Gui, and W. G. J. Halfond, “An empirical study of the energy consumption of android applications,” in *Proc. of ICSME*, 2014, pp. 121–130.
- [14] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, “Measuring user confidence in smartphone security and privacy,” in *Proc. of the Symposium on Usable Privacy and Security*, 2012.
- [15] WADL - Web Application Description Language. <http://www.w3.org/Submission/wadl/>.
- [16] RAML - RESTful API Modeling Language. <http://raml.org/>.
- [17] APIs.guru - Wikipedia for Web APIs. <https://apis.guru/>.
- [18] IBM API Connect. <https://developer.ibm.com/apiconnect/>.
- [19] Swagger Core Library. <https://github.com/swagger-api/swagger-core>. [20] swagger-jsdoc. <https://github.com/Surnet/swagger-jsdoc>.
- [21] S.M.Sohan,C.Anslow,andF.Maurer,“SpyREST: AutomatedRESTful API Documentation Using an HTTP Proxy Server,” in *Proc. of ASE*, 2015, pp. 271–276.
- [22] A. Feldthaus, M. Schařer, M. Sridharan, J. Dolby, and F. Tip, “Efficient construction of approximate call graphs for JavaScript IDE services,” in *Proc. of ICSE*, 2013, pp. 752–761.
- [23] M. Schäfer, M. Sridharan, J. Dolby, and F. Tip, “Dynamic determinacy analysis,” in *Proc. of PLDI*, 2013, pp. 165–174.
- [24] E.Andreasenand A. Møller,“Determinacy in static analysis for jQuery,” in *Proc. of OOPSLA*, 2014, pp. 17–31.
- [25] Y. Ko, H. Lee, J. Dolby, and S. Ryu, “Practically tunable static analysis framework for large-scale JavaScript applications,” in *Proc. of ASE*, 2015, pp. 541–551.
- [26] UsageStatisticsofJavaScriptLibrariesforWebsites,August2016.<https://w3techs.com/technologies/overview/javascript/library/all/>. [27] Atom Editor. <https://atom.io/>.

