RestTestGen

A Fully Automated Black-box Approach to Automatically Generate Test Scenarios for RESTful APIs

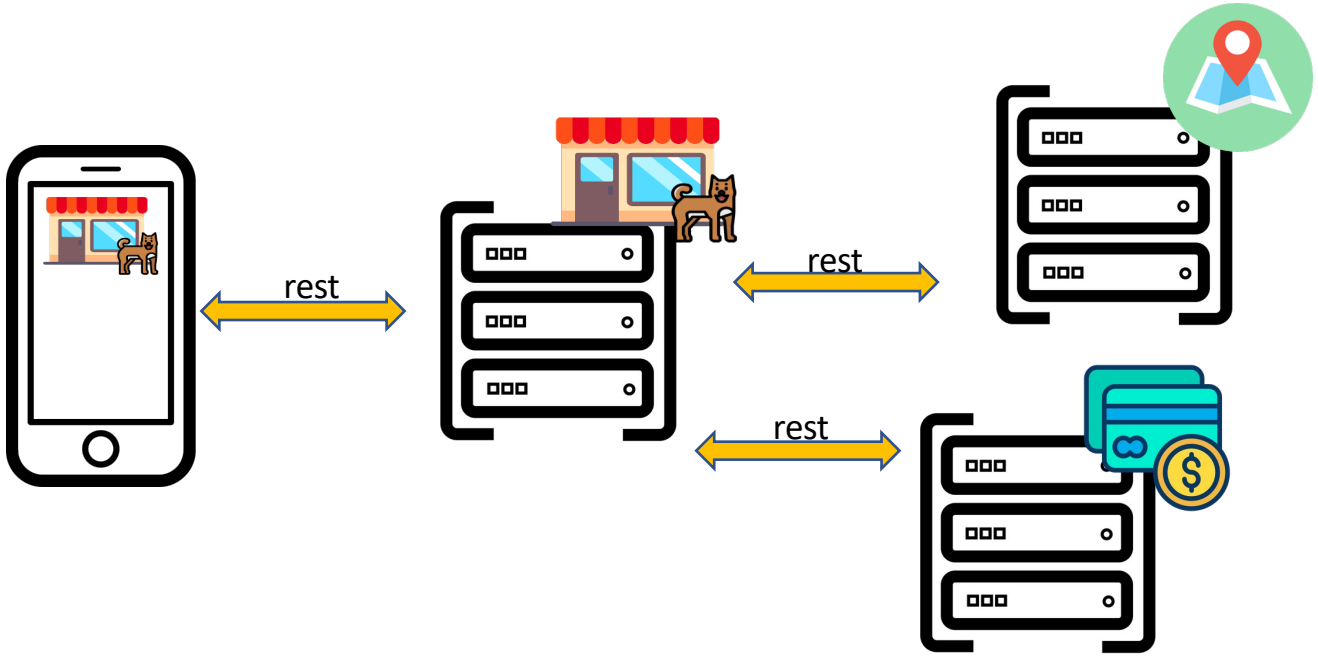
Mariano Ceccato

In collaboration with: Davide Corrandi, Michele Pasqua, Amedeo Zampieri

Department of Computer Science, University of Verona

mariano.ceccato@univr.it

REST APIs



Background – REST API

PetStore
https://petstore.swagger.io/v1

`post /pets`	Create new pet entry
`get /pets`	Read pet data
`put /pets`	Modify pet data
`delete /pets`	Remove pet data

Background – OpenAPI/Swagger

```yaml
openapi: "3.0.0"
info:
  version: 1.0.0
  title: Swagger Petstore
  license:
    name: MIT
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets/{petId}:
    get:
      summary: Info for a specific pet
      operationId: getPetById
      tags:
        - pets
      parameters:
        - name: petId
          in: path
          required: true
          description: The id of the pet to retrieve
          schema:
            type: string
      responses:
        '200':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pet"
        default:
          description: unexpected error
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Error"
```

URL to reach the API

Operations

HTTP methods

Input data & schema

Status code
Output data
Schema

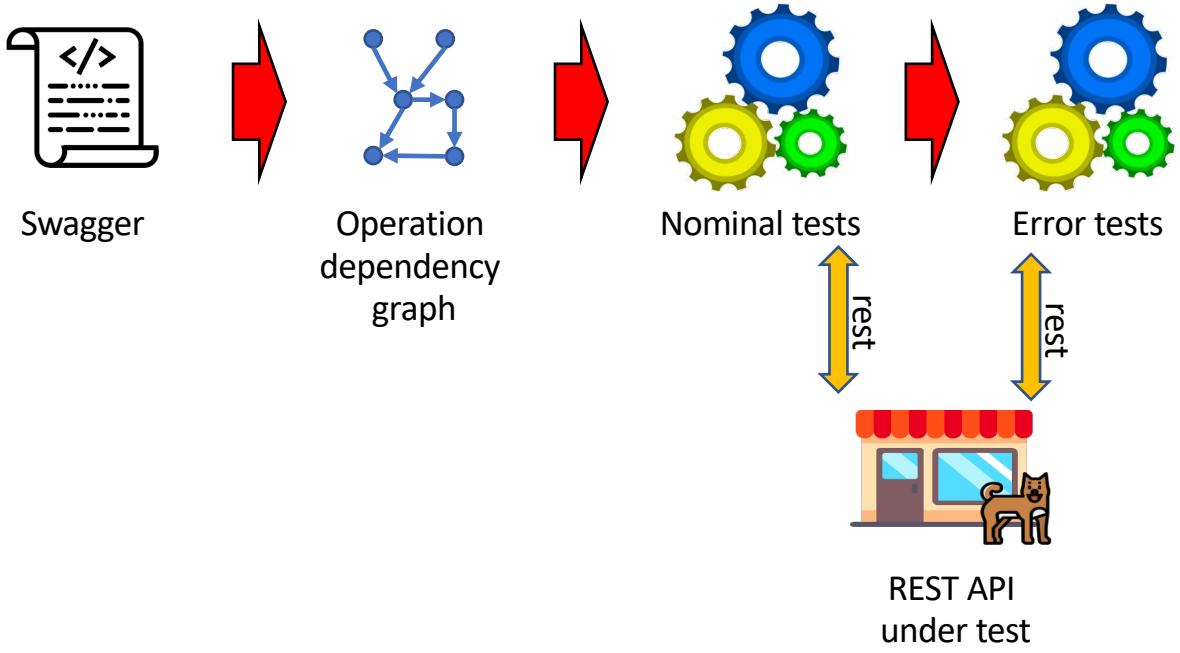
```yaml
components:
  schemas:
    Pet:
      type: object
      required:
        - id
        - name
      properties:
        id:
          type: integer
          format: int64
        name:
          type: string
        tag:
          type: string
```

Problem definition

- The number of REST APIs grows larger and larger
- Public REST APIs are used by many users and companies
  - might have error handling problems and implementation inconsistencies
- Source code analysis might be difficult due to many diversified distributed components (e.g., dynamically deployed, micro services)
- Often an API interface definition is available to clients
  - OpenAPI/Swagger

Solution:
Black-box testing of REST APIs based on their interface:
- Input/output format & domain
- Dependency among operations
- HTTP response status code

Approach overview



Producer-consumer dependencies

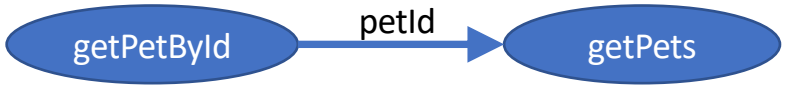
```
/pets:
   get:
     summary: List all pets
     operationId: getPets
     tags:
       - pets
     responses:
       '200':
         description: PetIds
         content:
           application/json:
             schema:
               type: array
               items:
                 type: object
                 properties:
                   petId:
                     type: integer
```

output

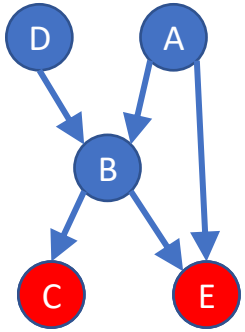
```
/pets/{petId}:
    get:
      summary: Info for a specific pet
      operationId: getPetById
      tags:
        - pets
      parameters:
        - name: petId
          in: path
          required: true
          schema:
            type: string
```

input

Case mismatch
petID, petid, petId
Id completion
/getPet ➡ Pet
pet.id ➡ petId
Stemming
pets ➡ pet



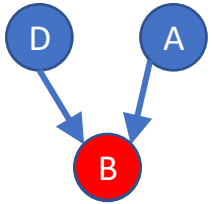
Operation Testing Order



- Leaf nodes are selected (no outgoing edges)
  - No input
  - Input is not available on operations output
- To maximize the likelihood of a successful test, resources might require to be in a certain status
- Leaf nodes are order based on the CRUD semantics

1. head
2. post
3. get
4. put/patch
5. delete

Operation Testing Order



- Tested operations are removed from the graph
- New operations become leaf nodes and can now be tested

The order in which operations are tested can not be precomputed, because it depends on what operations we succeed in testing

Input Value Generation

- Based on response dictionary
  - Map (name→values) of data observed at testing time, while testing previous operations
    - Exact name match — petId ✓ petId
    - Concatenation of object + field — pet.id ✓ petId
    - Name edit distance < threshold — petsId ✓ petId
    - Key is a substring — myPetId ✓ petId

- Based on swagger definition
  - Enum, example, default values
  - Random values (compatible with constraints)

HTTP Status Code Oracle

- 2xx means correct execution
  - 200: ok
  - 201: successful resource creation

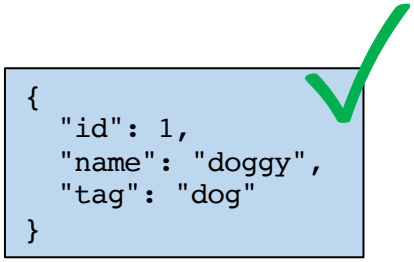
- 4xx means error that is correctly handled
  - 400: bad request
  - 404: not found

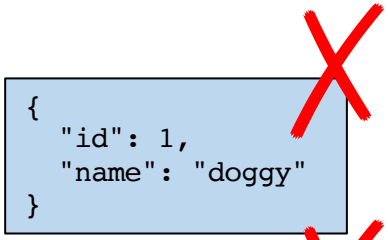
- 5xx means error
  - 500: server crash

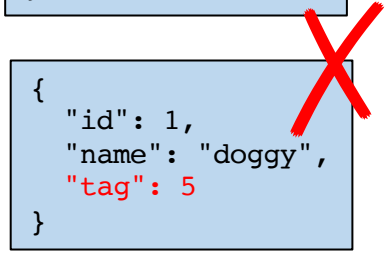
Schema Validation Oracle

```yaml
responses:
  '200':
    description: Expected response to a valid request
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/Pet"
```

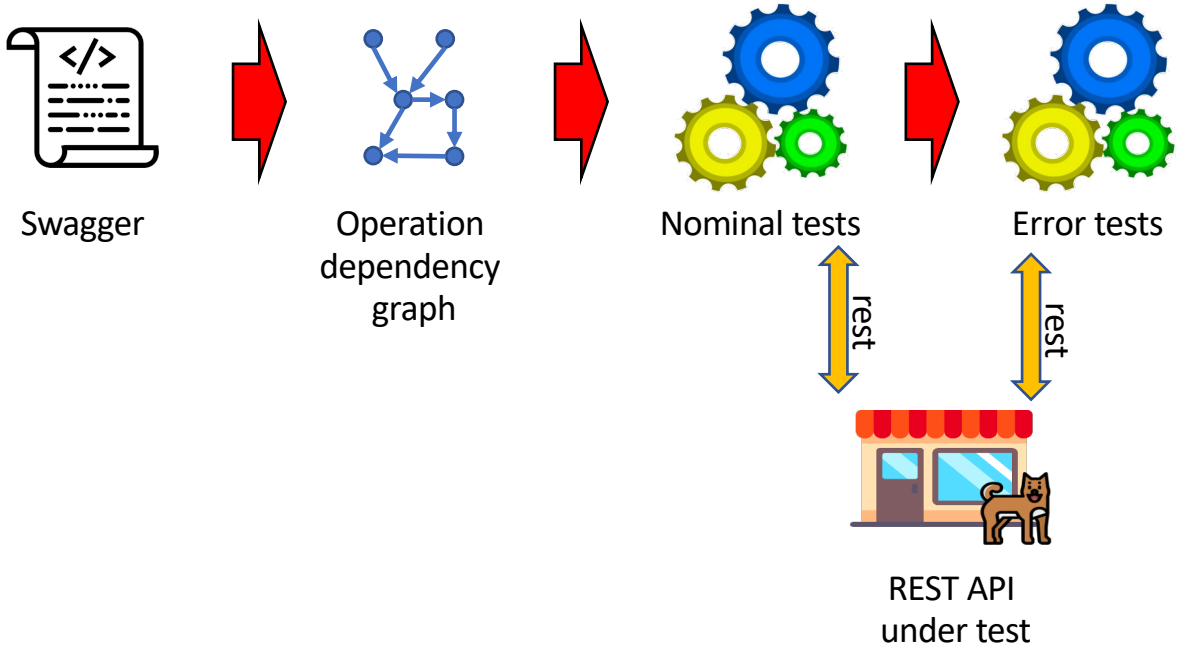
```yaml
components:
  schemas:
    Pet:
      type: object
      required:
        - id
        - name
      properties:
        id:
          type: integer
          format: int64
        name:
          type: string
        tag:
          type: string
```





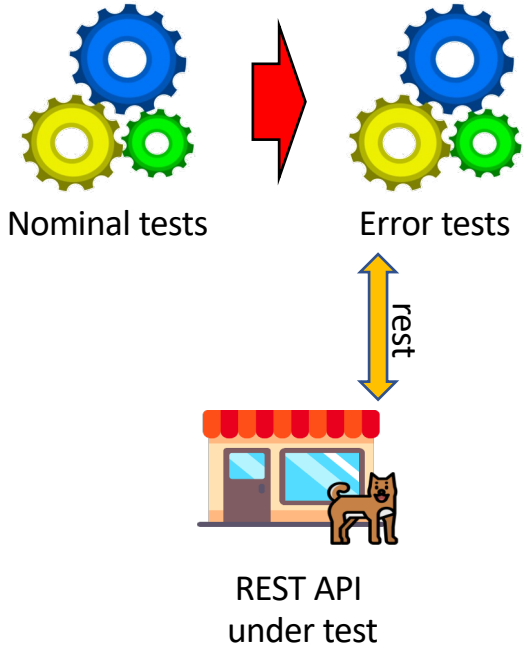


Approach overview



Testing of Error Cases

- Analyses how an API behaves when it is given wrong input data
- Mutation operators
  - Remove a required input field
  - Change field type
  - Change field value



HTTP Status Code Oracle

- 2xx means correct execution
  - 200: ok
  - 201: successful resource creation

- 4xx means error that is correctly handled
  - 400: bad request
  - 404: not found

- 5xx means error
  - 500: server crash

Experimental Validation

Research questions

- Is the Nominal Tester module effective in generating test cases?
- Is the Error Tester module effective in generating test cases?

Case studies

- 87 REST APIs listed in the website API.guru (2.6k operations)
  - Filtering out those with authentication or not responding

Procedure

- Nominal tester for 10 minutes per REST API
- Test cases with 2xx status code are mutated
- Error tester for 10 minutes per REST API
  - N_{fuzz}=5
  - Response dictionary threshold=1

Results

	APIs	Operations
Total	**87**	**2,612**
Status code 2xx	62	625
Status code 5xx	20	151
Validation error	66	1,733

Mutation operator	Mutants	Status code 2xx	Status code 5xx
Missing required	459	283	7
Wrong input type	707	511	16
Constraint violation	119	68	11
Total	**1,285**	**864**	**23**

[1] Corradini, D., Zampieri, A., Pasqua, M., Viglianisi, E., Dallago, M., & Ceccato, M. (2022). Automated black-box testing of nominal and error scenarios in RESTful APIs. Software Testing, Verification and Reliability, e1808.

Alternative approaches

RestTestGen [5]

- Operation Dependency Graph

RESTler [6]

- Full enumeration of sequences

bBOXRT [7]

- Large collection of mutation operators

RESTest [8]

- Inter-parameter dependencies

[2] E. Viglianisi, M. Dallago, and M. Ceccato, “RESTTESTGEN: Automated black-box testing of RESTful APIs,” in 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST), 2020, pp. 142–152

[3] V. Atlidakis, P. Godefroid, and M. Polishchuk, “RESTler: Stateful REST API fuzzing,” in Proceedings of the 41st International Conference on Software Engineering, ser. ICSE ’19. Piscataway, NJ, USA: IEEE Press, 2019, pp. 748–758.

[4] N. Laranjeiro, J. Agnelo, and J. Bernardino, “A black box tool for robustness testing of REST services,” IEEE Access, vol. 9, pp. 24 738–24 754, 2021.

[5] A. Martin-Lopez, S. Segura, and A. Ruiz-Cortés, “RESTest: Black-box constraint-based testing of RESTful web APIs,” in Service-Oriented Computing - 18th International Conference, ICSOC 2020, Dubai, United Arab Emirates, December 14-17, 2020, Proceedings, ser. Lecture Notes in Computer Science, E. Kafeza, B. Benatallah, F. Martinelli, H. Hacid, A. Bouguettaya, and H. Motahari, Eds., vol. 12571. Springer, 2020, pp. 459–475.

Coverage metrics

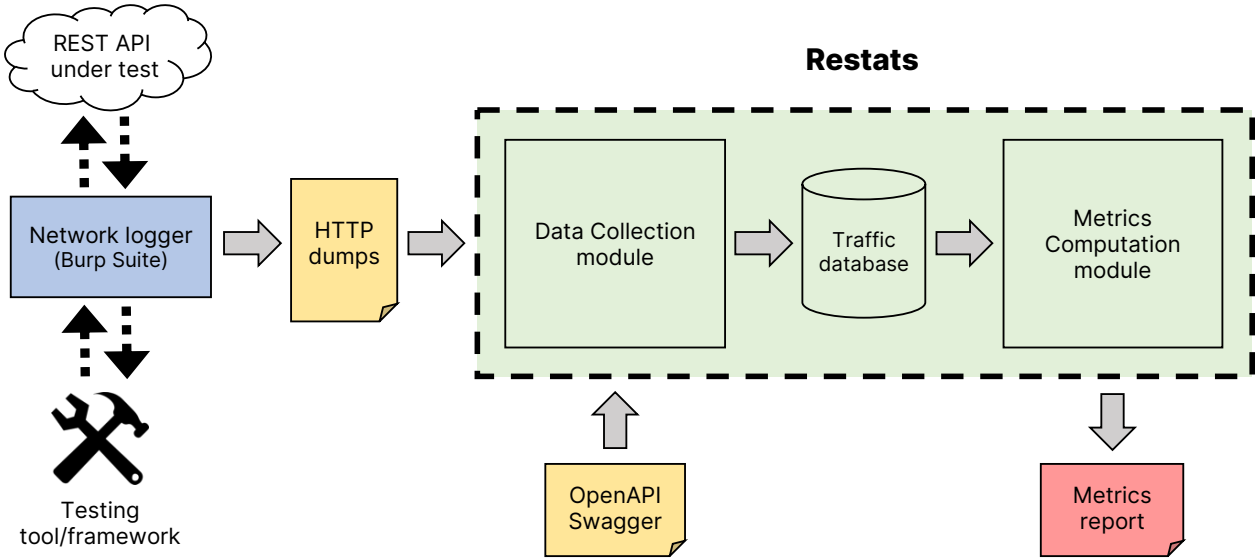
Input coverage metrics

- Path coverage
- Operation coverage
- Parameter coverage
- Parameter value coverage
- Request content-type coverage

Output coverage metrics

- Status code class coverage
- Status code coverage
- Response content-type coverage

RESTAT



Metric tool: RESTAT https://github.com/SeUniVr/restats

[6] Corradini, D., Zampieri, A., Pasqua, M., & Ceccato, M. (2021, September). Restats: A test coverage tool for RESTful APIs. In 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 594-598). IEEE.

REST APIs comparison case studies

REST APIs

- whose state can be reset after each test session
- that comes with an OpenAPI specification
- that are representative of real-world REST APIs

Case Study	Language	Framework	Endpoints	Operations	# of lines
01-Slim	PHP	Slim	9	18	8,566
02-Airline	Java	Spring Boot	12	30	3,859
03-Streaming	Java	Spring Boot	5	5	1,780
04-Petclinic	Java	Spring Boot	17	47	8,550
05-Toggle	ASP.NET	.NET Core	8	16	2,363
06-Problems	Java	Spring Boot	5	9	2,174
07-Products	Java	Spring Boot	6	14	3,451
08-Widgets	Go	-	4	14	1,370
09-Safrs	Python	Flask	6	18	2,787
10-Realworld	PHP	Laravel	11	19	5,278
11-Crud	Node.js	Express	1	4	5,106
12-Order	PHP	Laravel	2	3	3,359
13-Users	TypeScript	Express	2	5	805
14-Scheduler	Node.js	Express	26	40	24,044

Research questions

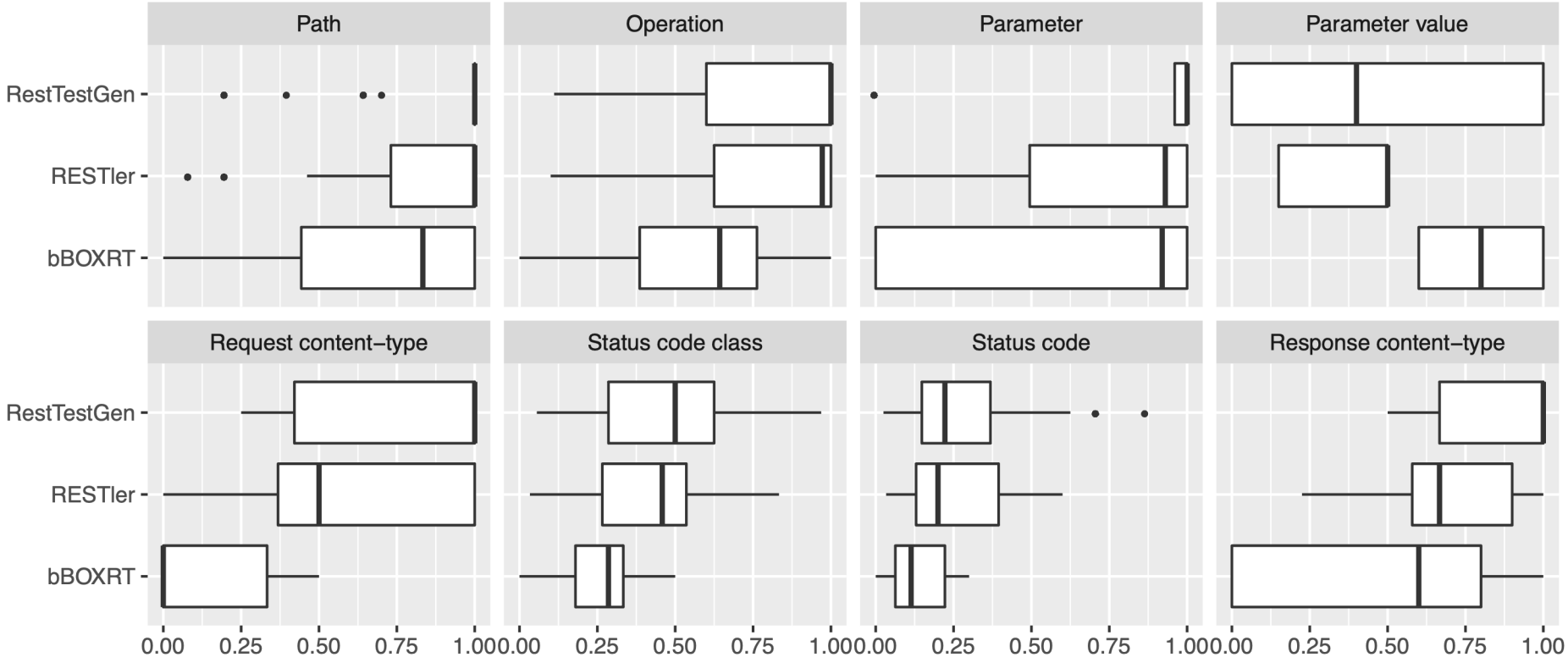
1. How *robust* are automated RESTful APIs test-case generation tools?

2. What is the *coverage* of the test suites emitted by automated RESTful APIs test-case generation tools?

Results: robustness

Case study	RestTestGen	RESTler	bBOXRT	RESTest
01-Slim	✓	✓	✗	✓
02-Airline	✗	✓	✗	✗
03-Streaming	✗	✓	✗	✗
04-Petclinic	✓	✓	✗	✗
05-Toggle	✓	✓	✓	✗
06-Problems	✓	✓	✗	✗
07-Products	✓	✓	✓	✗
08-Widgets	✓	✓	✓	✓
09-Safrs	✓	✓	✓	✗
10-Realworld	✓	✓	✓	✗
11-Crud	✓	✓	✓	✗
12-Order	✓	✓	✓	✗
13-Users	✓	✓	✓	✗
14-Scheduler	✗	✓	✗	✗
Total	11	14	8	2

Results: coverage



Results: coverage

Coverage metric	RestTestGen	RESTler	bBOXRT	Draw
Path	1	0	0	7
Operation	1	3	0	4
Parameter	1	0	0	4
Parameter value	1	0	2	0
Req. content-type	2	1	0	4
Status code class	4	4	0	0
Status code	5	3	0	0
Resp. content-type	3	2	0	2

Number case studies for which a tool performed better than the others.

[7] Corradini, D., Zampieri, A., Pasqua, M., & Ceccato, M. (2021, September). Empirical comparison of black-box test case generation tools for restful apis. In 2021 IEEE 21st International Working Conference on Source Code Analysis and Manipulation (SCAM) (pp. 226-236). IEEE.

Considerations

- One sequence Vs many sequences
  - RestTestGen when time and resources are limited
  - RESTler when a lot of time and resources are available
- bBOXRT is effective in fault detection
  - High score for parameter value metric
- RESTest is still not mature for real-world REST APIs
  - Inter-parameter dependencies can be helpful in testing

From tool to framework

Core components

- OpenAPI Parser
- Operation
- Data template
- Test sequence
- Test interaction
- Test runner
- Operation dependency graph
- Dictionary

Extensible components

- Operation ordering
- Input Value provider
- Mutation operator
- Test case writer
- Strategy
  - Nominal Testing Strategy
  - Error Testing Strategy

https://github.com/SeUniVr/RestTestGen

Developers

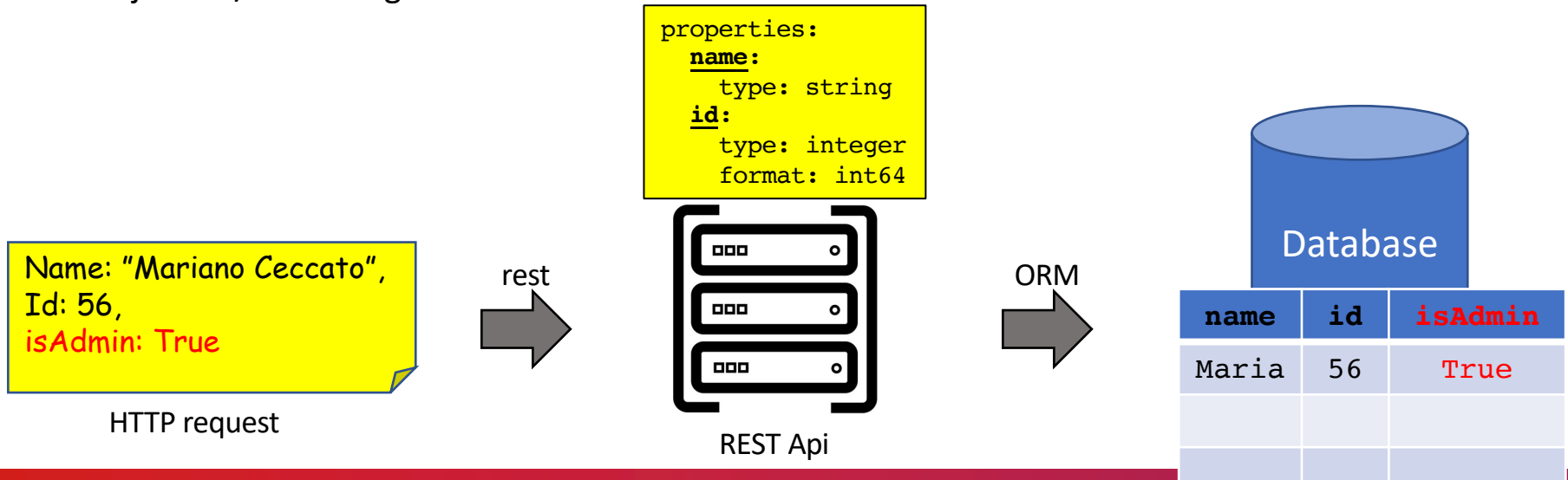
Stakeholders

Researchers

Ongoing research

Mass assignment vulnerability

- Input parameters that are not specified in the OpenAPI, but accepted by the implementation
  - Read-only parameters
    - isAdmin flag?
  - Common (internal data)
    - Timestamp?
- Automatic binding between input and persistency backend (e.g., database table)
- Injection/overriding of data



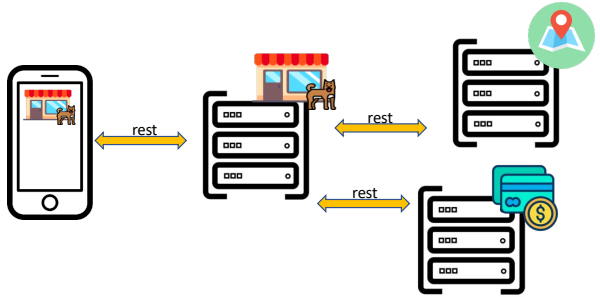
Broken access control

- Features and resources only available to certain users
  - Is access check enforced only at client-side?
  - Does the API assume client does not know the resource IDs?

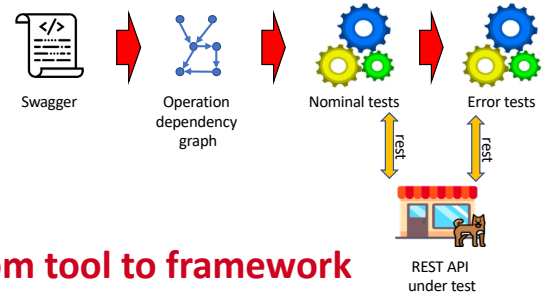
	news.doc	photo.png	fun.com
Admin	read	view edit	view
Bob	read write	view edit	view modify
Charlie			
Dave		view	

Conclusion

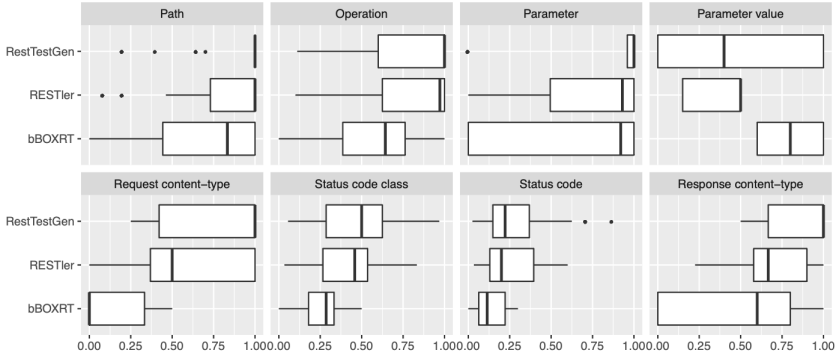
REST APIs



Approach overview



Results: coverage



From tool to framework

Core components
- OpenAPI Parser
- Operation
- Data template
- Test sequence
- Test interaction
- Test runner
- Operation dependency graph
- Dictionary

Extensible components
- Operation ordering
- Input Value provider
- Mutation operator
- Test case writer
- Strategy
  - Nominal Testing Strategy
  - Error Testing Strategy

https://github.com/SeUniVr/RestTestGen

Developers Stakeholders Researchers