

Improving the implementation of microservice-based systems with static code analysis

aka.

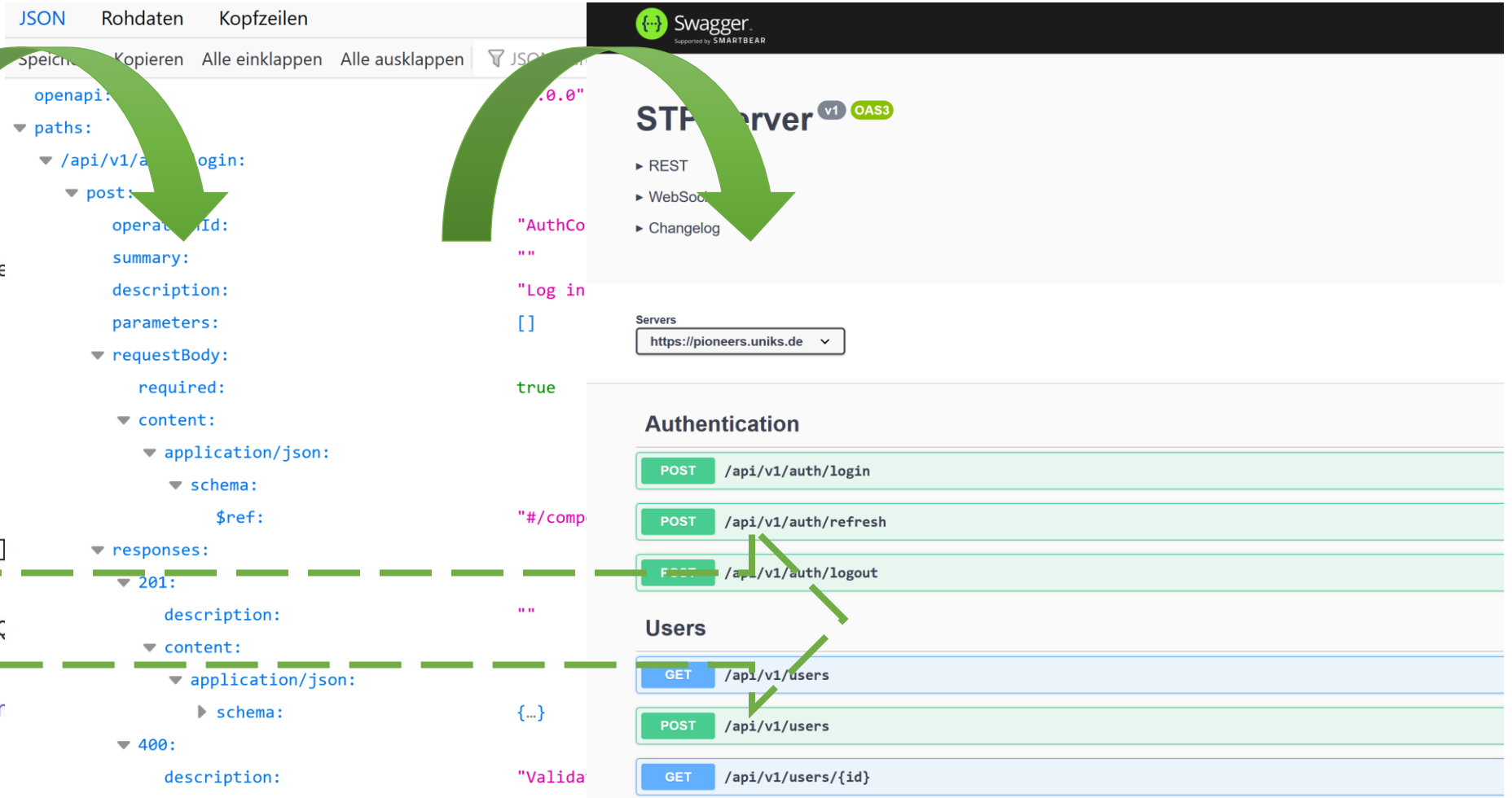
Improve the IDE support for developers of REST-based microservices to
keep API specification and implementation consistent

But why?

```

30 @Controller('users')
31 @ApiTags('Users')
32 @Validated()
33 @Throttled()
34 export class UserController {
35     constructor(
36         private userService: UserService
37     ) {
38     }
39
40     @Get()
41     @Auth()
42     @ApiOperation({ description:
43     @ApiResponse({ type: [User]
44     async getUsers(
45         @Query() { status, ids }: {
46     ): Promise<User[]> {
47         return this.userService.fir
48     }
49

```



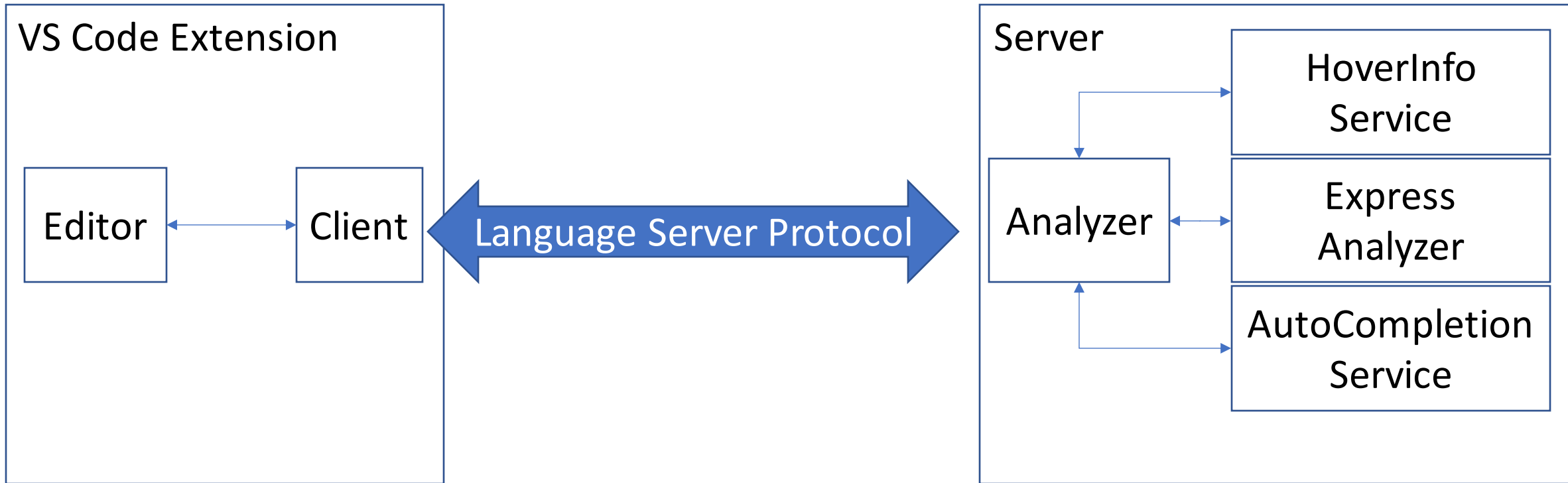
The screenshot shows the Swagger UI for 'STP Server v1 OAS3'. The 'paths' section is expanded to show a POST endpoint for '/api/v1/auth/login'. The 'Authentication' section lists endpoints for login, refresh, and logout. The 'Users' section lists endpoints for listing users and getting a user by ID. Green arrows indicate the mapping from the code to the Swagger UI elements.

The what?

- A plugin for Visual Studio Code integrated with the language server protocol
- Automatically compares API specifications with implementation
 - Type Checking
 - Endpoint information on Hover
 - Enable Auto completion of routes
 - Jump to definition (from a frontend, if projects are a monorepo env.)

DEMONO

How?



You can't be the first one, right?

- Formal approaches
 - Formal Verification of Stateful Services with REST APIs Using Event-B [1]
 - SafeRESTScript: Statically Checking REST API Consumers [2]
- Practical approaches (without IDE integration)
 - Statically Checking Web API Requests in JavaScript [3]
 - Swagger-based jQuery Ajax Validation [4]
- Practical approaches (with IDE integration)
 - Opportunities in Software Engineering Research for Web API Consumption [5]

What's next?

- Replace the configuration file (.siarc) with a Swagger integration
 - Supports both OpenAPI version 2 and version 3
- Publishing the extension at the Visual Studio Code marketplace
- Evaluates the usage of the extension
 - With students from lectures
 - With a user base whose downloaded the extension from the marketplace

Literature

- [1] <https://ieeexplore.ieee.org/abstract/document/8456341>
- [2] <https://arxiv.org/abs/2007.08048>
- [3] <https://ieeexplore.ieee.org/document/7985666>
- [4] <https://ieeexplore.ieee.org/document/8666542>
- [5] <https://ieeexplore.ieee.org/document/7965485>

Autiores

Software Engineering Research Group
Kassel University

Sebastian Copei
sco@uni-kassel.de

Kassel University

Maximilian Schreiter
maximilian.schreiter@t-online.de

Software Engineering Research Group
Kassel University

Prof. Dr. Albert Zündorf
zuendorf@uni-kassel.de