

CHAOS IM API DSCHUNDEL

Design-Guidelines als Orientierungshilfe

Constanze Klaar, Benjamin Klatt







450+ microservices

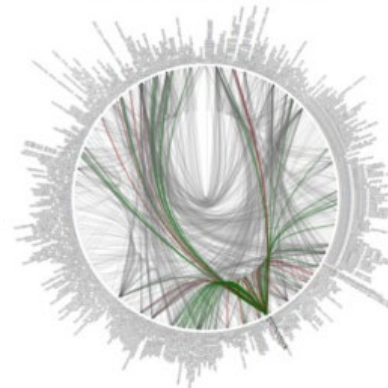


NETFLIX

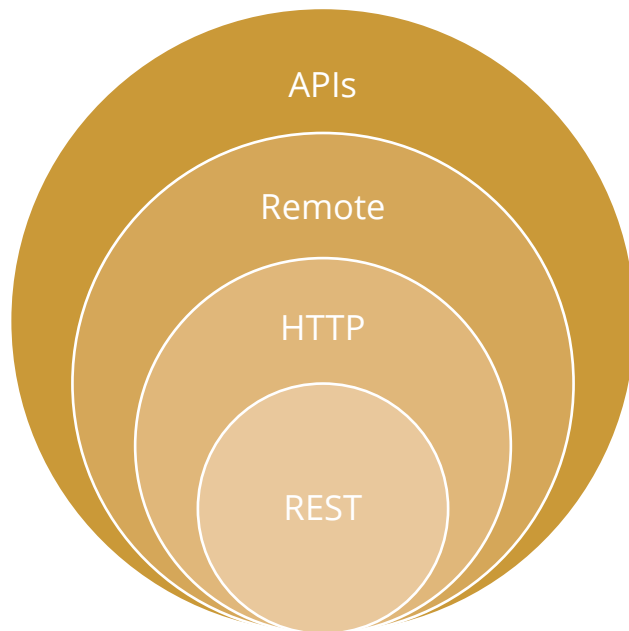
500+ microservices



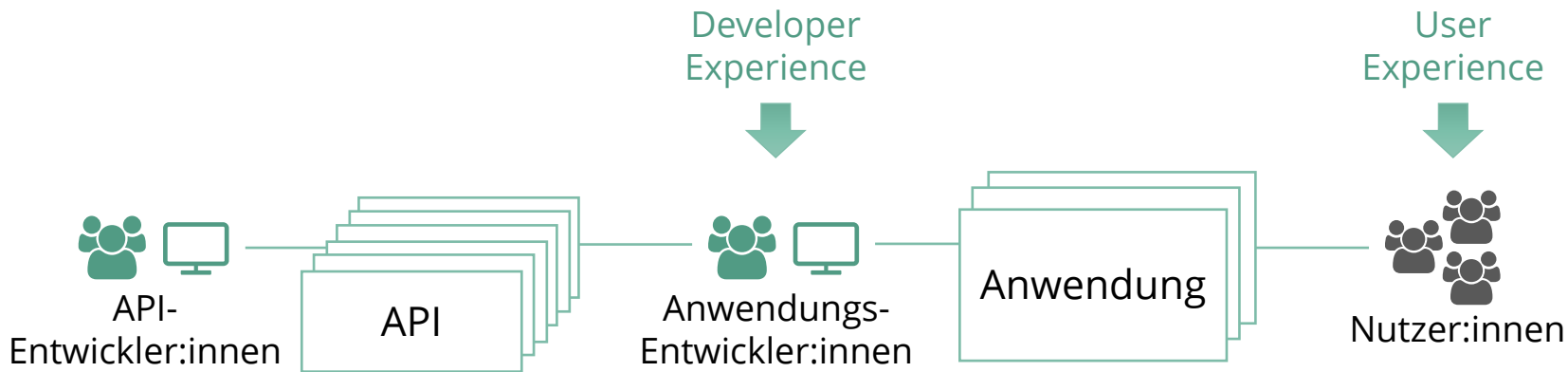
500+ microservices

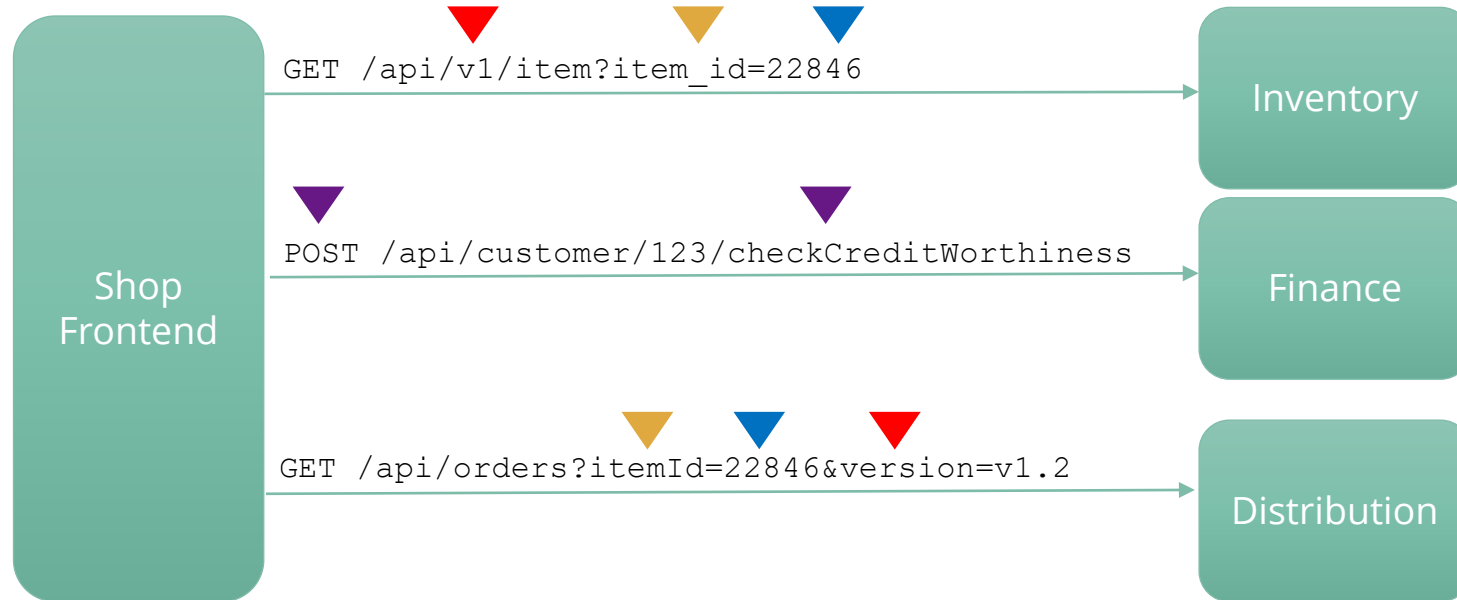


FOKUS FÜR HEUTE



USER EXPERIENCE – DEVELOPER EXPERIENCE





▼ = Inkonsistenzen / Unstimmigkeiten

NORMAN DOORS

DRÜCKEN ODER ZIEHEN?

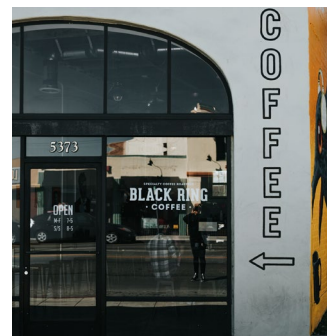
STRATEGIEN ZUR ORIENTIERUNG



Infrastruktur

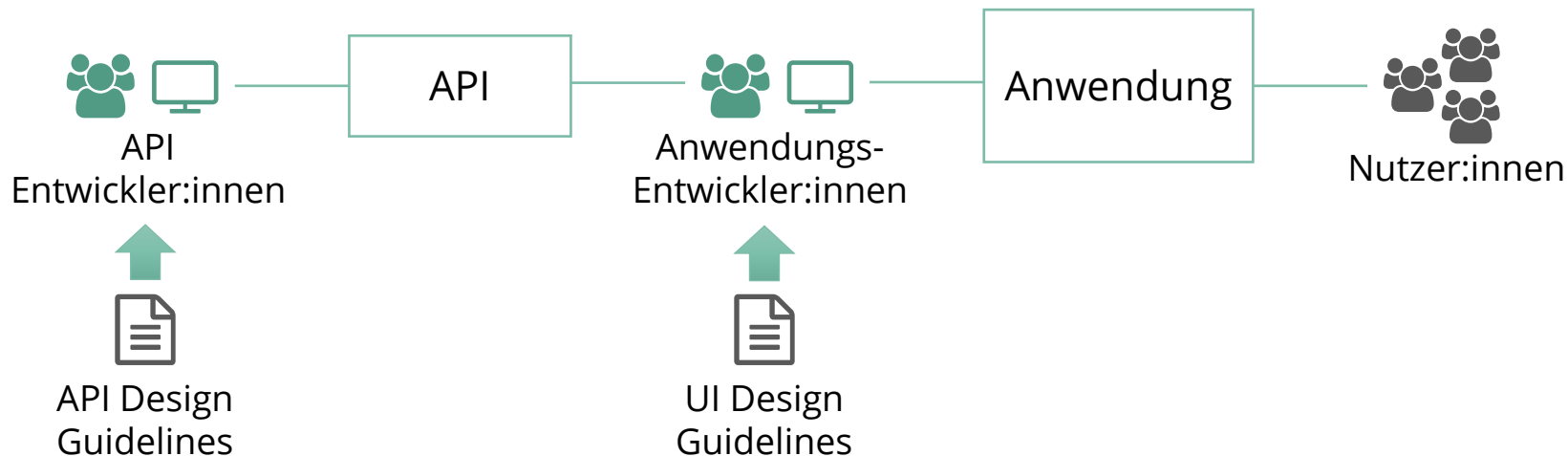


Support



Guidelines

USER EXPERIENCE – DEVELOPER EXPERIENCE



ÖFFENTLICHE API DESIGN GUIDELINES

27 Kategorien	Anzahl (von 32)
Status Codes	30
Response Format	29
Standard Methods	29
Naming	28
Versioning	28
Pagination	24
URL/URI Structures	24
Error Response	22
Filter	17
HTTP Field/Header	15
Security	15
Backwards Compatibility	13
Naming Resources	13
Caching	12

Documentation	12
URI Field	12
Sorting	11
Action Resources	10
CORS	9
Long Running Ops	7
Rate Limiting	6
GZIP Compression	5
Metadata	4
Naming Collections	4
Custom Methods	2
Empty Responses	2
Rules for API Users	2



zalando



HAUFE.



...

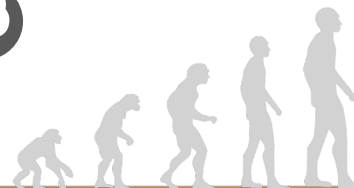
BEISPIEL: HTTP STATUS CODES

200er	300er	400er	500er
Success	Moved	Bad Request	Server Error
201 created 202 accepted ...	303 see other 304 not modified ...	401 unauthorized 429 too many req ...	501 not implemented 503 service unavail. ...

BEISPIEL: VERSIONIERUNG

Versionierungsmöglichkeiten

- URI Path
(www.example.com/api/**1**/products)
- Query Parameter
(www.example.com/api/products?**version=1**)
- Custom Header
(curl -H "**Accepts-version: 1.0**")
- Content Negotiation
(curl -H "Accept: application/vnd.xml.device+json; **version=1**")



BEISPIEL: DATUMSFORMATE

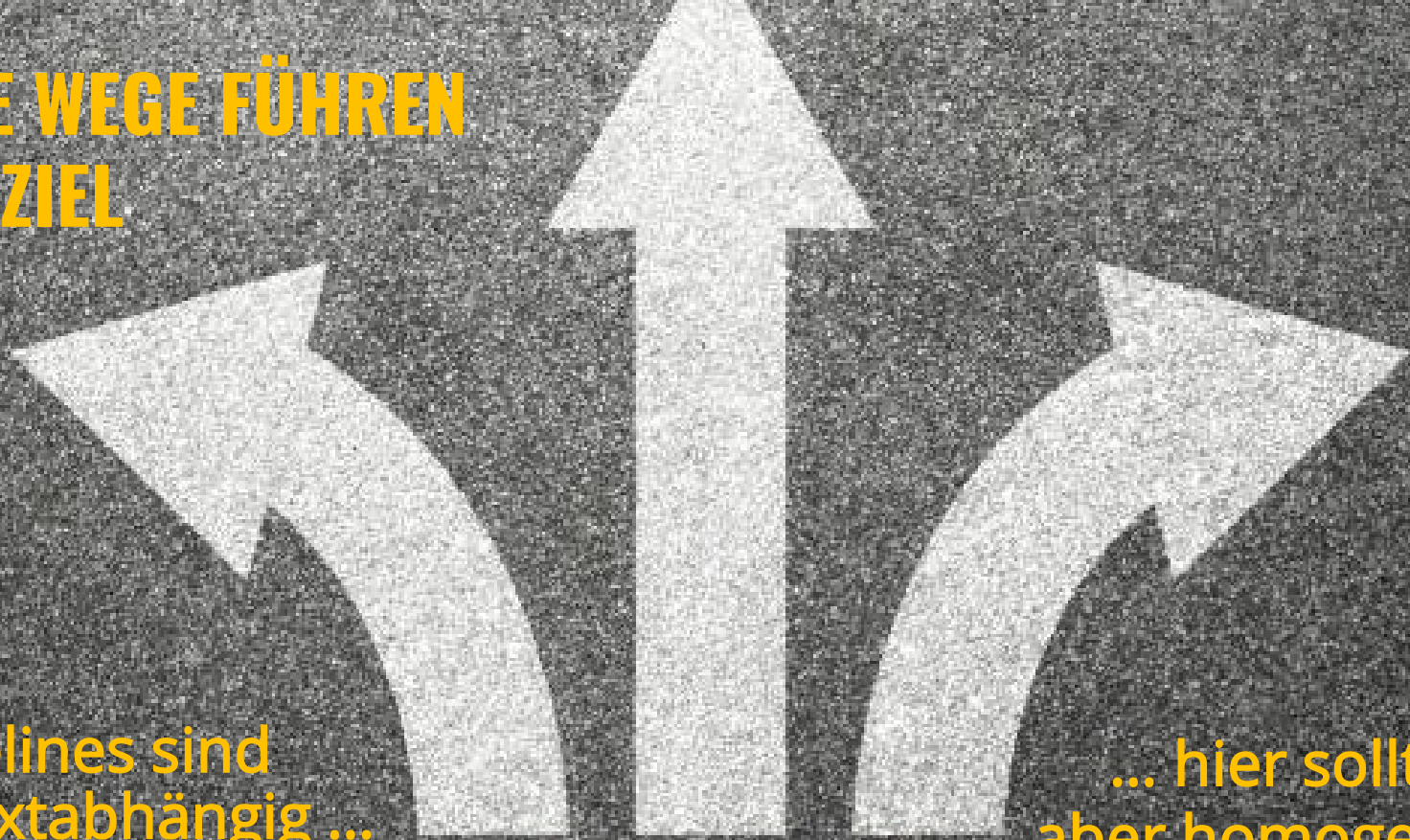
Parameter mit Datumsangaben

```
"shipDate": {  
  "type": "string",  
  "format": "date-time"  
},
```

Was bedeutet date-time?

- RFC 5322: Tue, 09 Feb 2023 11:00:00 GMT
- RFC 3339: 2023-02-09T11:00:00-08:00
- ISO-8601: 2023-02-09T11:00:00+01:00

VIELE WEGE FÜHREN ZUM ZIEL



Guidelines sind
kontextabhängig ...

... hier sollten sie
aber homogen sein

API GUIDELINE EINHALTUNG



653 Open API
Spezifikationen*



38 Regeln
Spectral Linter



220 errors
58817 warnings
12789 infos

EINFÜHRUNG VON API DESIGN GUIDELINES

DO'S AND DONT'S

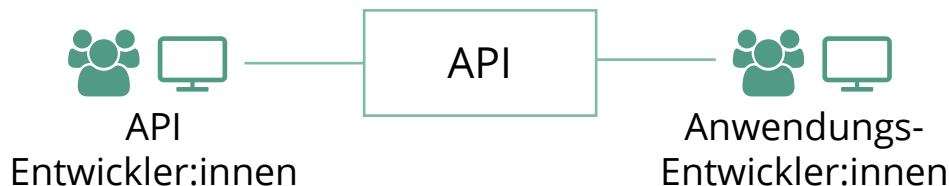


API Design Guidelines:

- die wenigsten API-Entwickler:innen lesen sie
- falls doch erinnern sie sich nicht
- oder schauen nicht nach Änderungen

- Phil Sturgeon*

MEHRWERT GUTER API DESIGN GUIDELINES



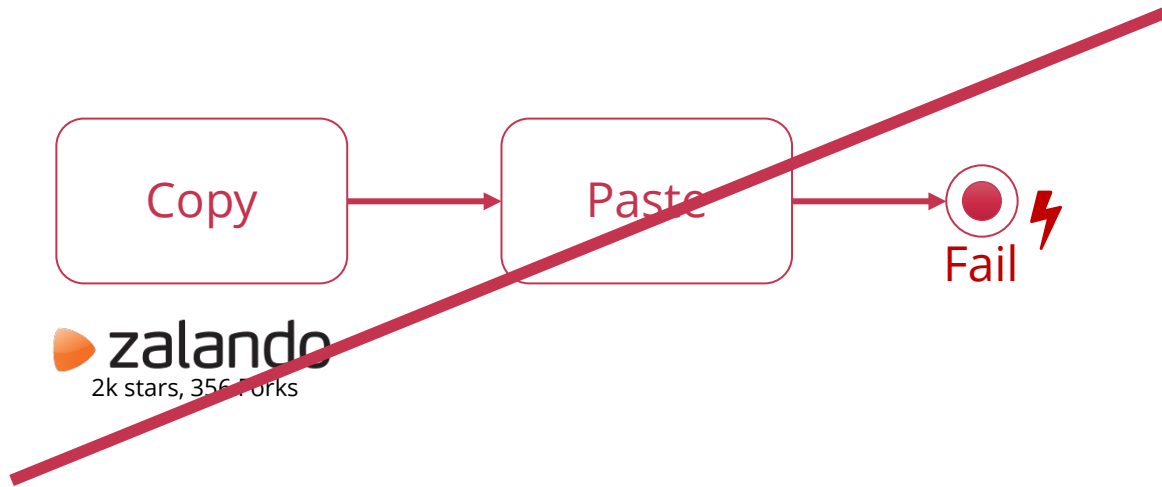
API-ENTWICKLER:INNEN

- Reduzieren API Design Aufwand
- Unterstützen API Reviews
- Fördern Tool Unterstützung

API-KONSUMENTEN

- Fördern API-Usability und -Developer Experience
- Konsistenz
 - Einfachheit
 - Lernbarkeit (bekannte Muster)
 - Fehlervermeidung
 - Produktivität

TIPPS ZUR EINFÜHRUNG



TIPPS ZUR EINFÜHRUNG

GUIDELINE GESTALTUNG

- Kompakt & einfach zu lesen
- REST Standard berücksichtigen
- (OpenAPI) Beispiele
- Problemorientiert
- Zielgruppenorientiert

VORGEHEN

- Mehrwert kommunizieren
- Verständnis für Mehraufwand
- Sensibilisierung für DX
- Nutzer Feedback einholen
- Kontinuität

TOOL SUPPORT

- Leichter Zugang (Online)
- API-First Unterstützung („Linter“)
- Code-First Unterstützung

REST-API DESIGN CHECK

Improve your API design by analyzing your OpenAPI specifications

[COPY&PASTE](#)[UPLOAD](#)[TEST-DATA](#)

```
1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "Simple API overview",
5      "version": "2.0.0"
6    },
7    "paths": {
8      "/": {
9        "get": {
10          "operationId": "listVersionsv2",
11          "summary": "List API versions",
12          "responses": {
13            "200": {
14              "description": "200 response",
15              "content": {
16                "application/json": {
17                  "examples": {
18                    "foo": {
19                      "value": {
20                        "versions": [
21                          {
22                            "status": "CURRENT",
23                            "updated": "2011-01-21T11:33:21Z",
24                            "id": "v2.0",
25                            "links": [
26                              {
27                                "href": "http://127.0.0.1:8774/v2/",
28                                "rel": "self"
29                              }
30                            ]
31                          }
32                        ]
33                      }
34                    }
35                  }
36                }
37              }
38            }
39          }
40        }
41      }
42    }
43  }
```

[FORMAT JSON](#)[ANALYZE >](#)

Made with  by viadee

<

viadee API Guide

FEEDBACKHOMERULESRESULTS

Overview

1:CopiedOasFile.json

Add OAS

Remove all

General Properties

Endpoint Properties

Rule Violations

API Title	Simple API overview
OAS Version	3.0.0
File Fomat	JSON
API Version	2.0.0
Endpoints	2
Methods	2

Status-Code	#Endpoints
200	2
203	1
300	1

Media-Types	#Endpoints
application/json	4

Severity

Rule

Description

Path

Position

	Should define a description for the API	Info "description" must be present and non-empty string.	info	Row 3 To 5
	kebab-case path segments	Path segments are restricted to ASCII kebab-case strings matching regex <code>^[a-z][a-z-0-9]*\$</code> . This does not apply to path parameters!	paths /	Row 8 To 56

Made with  by [viadee](#)

CODE-FIRST UNTERSTÜTZUNG

Feedback bereits bei der Implementierung

```
@RequestMapping("/api/Users")
@Tag(name = "User API", description = "User Management API")
interface UserApi {

    @GetMapping
    @ResponseStatus(code = HttpStatus.OK)
    List<User> findAll();

    @GetMapping("/{id}")
    @ResponseStatus(code = HttpStatus.OK)
    User findById(@PathVariable String id);

    @PostMapping
    @ResponseStatus(code = HttpStatus.CREATED)
    User save(@RequestBody User user);

    @PutMapping("/{id}")
    @ResponseStatus(code = HttpStatus.OK)
    User update(@PathVariable String id, @RequestBody User user);

    @DeleteMapping("/{id}")
    @ResponseStatus(code = HttpStatus.NO_CONTENT)
    void delete(@PathVariable String id);
}
```


lower-
kebab-case

Generierung

```
{
  "openapi" : "3.0.1",
  "info" : {...},
  "servers" : [ ...],
  "tags" : [ {
    "name" : "User API",
    "description" : "User Management API"
  } ],
  "paths" : {
    "/api/Users/{id}" : {
      "get" : {
        "tags" : [ "User API" ],
        "operationId" : "findById",
        "parameters" : [ {
          "name" : "id",
          "in" : "path",
          "required" : true,
          "schema" : {
            "type" : "string"
          }
        } ],
        "responses" : {
          "404" : {
            "description" : "Not Found",
            "content" : {

```

APIUNIT

Maven

```
<dependency>
  <groupId>de.viadee</groupId>
  <artifactId>apiunit</artifactId>
  <version>0.1.0</version>
</dependency>
```

Unit Test

```
class ApiGuidelineTest {
    @Test
    public void checkApiTests() {
        ApiUnitTest myTest = new ApiUnitTest("com.example.demo.controller");
        myTest.addRule(new CollectionIdsMustBeLowerCamelCase());

        ApiUnitTestResultObject testResult = myTest.check();
        assertEquals(0, testResult.getViolationCounter());
    }
}
```

Test Ergebnis

```
ApiUnitTestResultObject{
violations (n=1)= [
  ApiRuleCheck{
    rule='CollectionIdsMustBeLowerCamelCase',
    location='com.example.demo.controller.UserApi.findAll()',
    description='Users is not lower camel case'
  }
]
```

WRAP UP

API Design Guidelines

- unterstützen die Developer Experience
- Strategie neben Infrastruktur und Support
- sind selbst als Produkt zu verstehen
(DX für API-Teams)

CHAOS IM API DSCHUNDEL

API DESIGN GUIDELINES FÜR EINE BESSERE DEVELOPER EXPERIENCE



Vielen Dank für Ihre Aufmerksamkeit
Dr. Benjamin Klatt
Constanze Klaar

✉ benjamin.klatt@viadee.de

in Benjamin Klatt

✉ constanze.klaar@viadee.de

in Constanze Klaar