
Atlassian Forge – Von der Idee zum Release

Daniel Kleissl

Disclaimer

- Kein:
 - React-Tutorial
 - In-Depth Forge-Tutorial
- Ein:
 - Forge-Tutorial für Einsteiger
 - Sprungbrett für die Entwicklung
 - Geteilter Erfahrungsschatz



Ressourcen

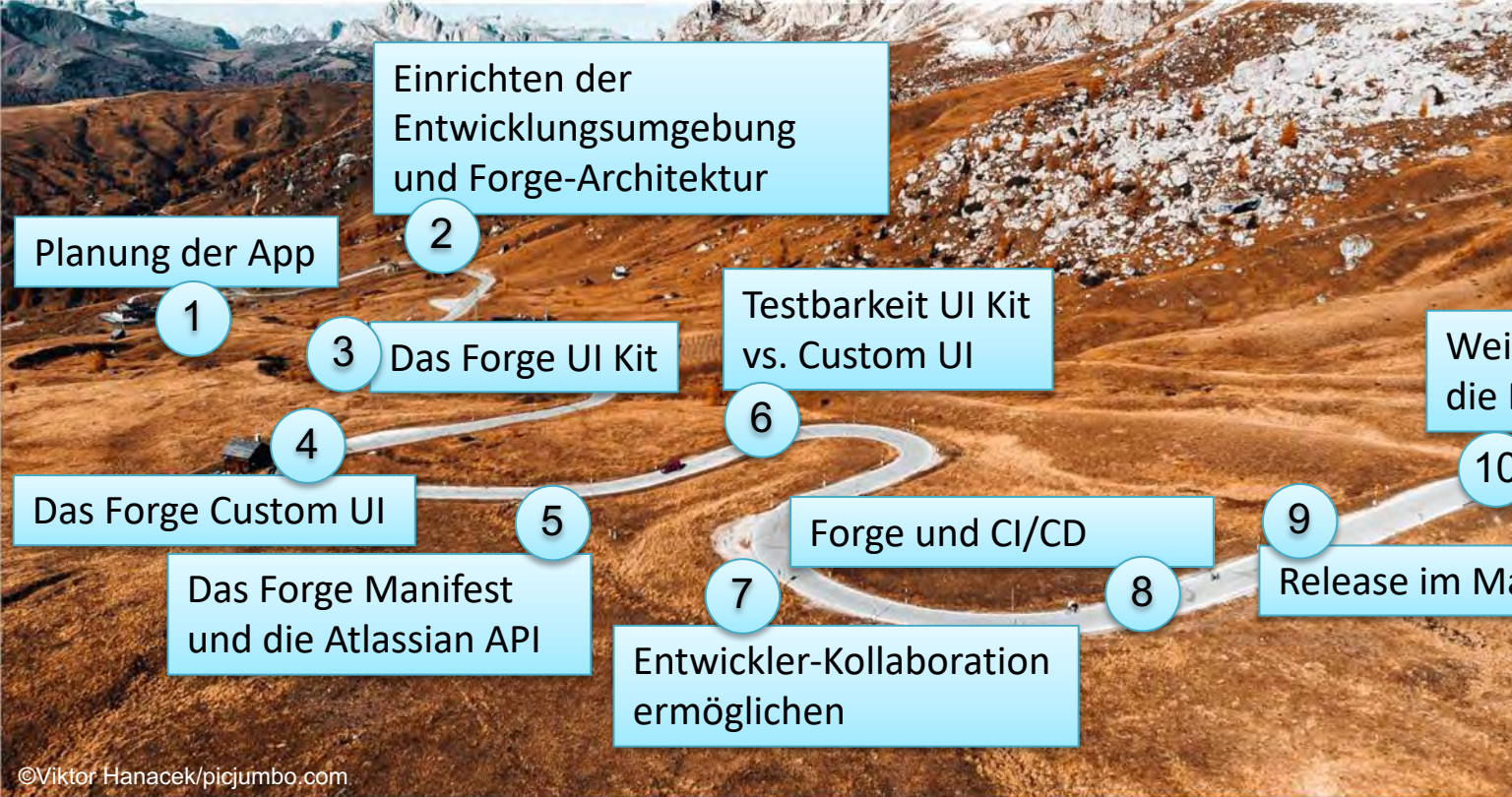
- Werden bereitgestellt:
 - Links
 - Code-Snippets



„Journey to Rome“ – Jira Risk Management

Berechnen Sie das Anforderungsrisiko und die Testintensität basierend auf 12 simplen Fragen und Antworten.

Weitere Informationen: www.razzfazz.io



©Viktor Hanacek/picjumbo.com

A photograph of a server room with rows of server racks. The racks are illuminated with green and yellow lights, and the floor is highly reflective, showing the lights from the racks.

Datacenter Apps

- Entwickelt mit Java
- Höhere Performance-Requirements
- Komplexer zu programmieren

On Prem vs. Cloud Apps

A photograph of a vast, white, fluffy cloud layer from an aerial perspective. Below the clouds, a small portion of a green landscape with some buildings is visible.

Cloud Apps

- Entwickelt mit JavaScript
- Über Cloud verteilt
- Einfacher zu programmieren

©Viktor Hanacek/picjumbo.com

Serverless app
development *platform*

Beginner

Für:

- Confluence Cloud 
- Jira Cloud 
- Jira Service Management Cloud 

Everyone

FaaS – *Function as a
Service*

Advanced

Was ist Forge?

©Viktor Hanacek/picjumbo.com

Welches Atlassian
Produkt?

Welche
Ansichten?

User?
API?
App?

App-
Funktionalität

Nur Anzeige?
Persistente
Datenänderung?
Intern vs. Extern?

Berechtigungen

Welcher Stack?
Welche Frameworks?



Jest



Mocha



Sinon



Chai



ChartJS

Etc.

Plane den Aufstieg

test task

Attach Create subtask Link issue Journey To Rome ...

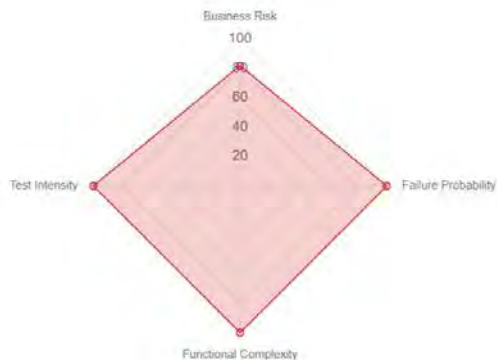
Description

Add a description...

Issue Panel

Journey To Rome DEV

Restart Risk Assessment



Custom field

To Do ▾

⚡ Actions ▾

Details

Assignee

👤 Unassigned

Assign to me

Reporter

👤 Daniel Kleissl

Labels

None

Risk evaluation DEV

Risk: A1

Complexity: HIGH

Test Intensity: 100

Values are computed

Priority

🟡 Medium

More fields Original esti... ▾

Umsetzung des UIs?

UI Kit

Custom UI

Welcher Workflow?

Was holt
sich **wo** die
Daten?

Welche Module?

Issue Panel
Custom Field
Dashboard

Plane den Aufstieg

©Viktor Hanacek/picjumbo.com



1. Plane den Aufstieg deiner App in die Cloud

- Überlege dir, was die App wirklich können soll
- Prüfe, ob Forge für deine Idee geeignet ist
- Planen ist wichtig, Machen ebenso

One
more
thing

Anlegen

- Atlassian Partner Account
- API Token
- Atlassian Developer Account
- Atlassian Produkt Instanz

Node 16+



Docker



VS Code (o.ä.)



Voraussetzungen

Wichtig!

Dependencies
updaten!

1. Forge CLI installieren
2. Login mit Forge
3. App anlegen

```
1 npm install -g @forge/cli
2
3 forge login
4
5 forge create
```

Beispiel-Apps auswählbar
Guter Anhaltspunkt für Entwicklung

Einrichten

©Engin_Akyurt/pixabay.com

App bekommt
eigene ID!

1. App registrieren

- Wenn schon vorhanden

2. App deployen

3. App installieren

Deploy und Install
nur für eigene App
möglich!

Deploy auf:
Develop
Staging
Production

```
1 forge register
2
3 forge deploy -e [environment]
4
5 forge install
```

Labels

None

Risk evaluation: DEV

Risk: C3

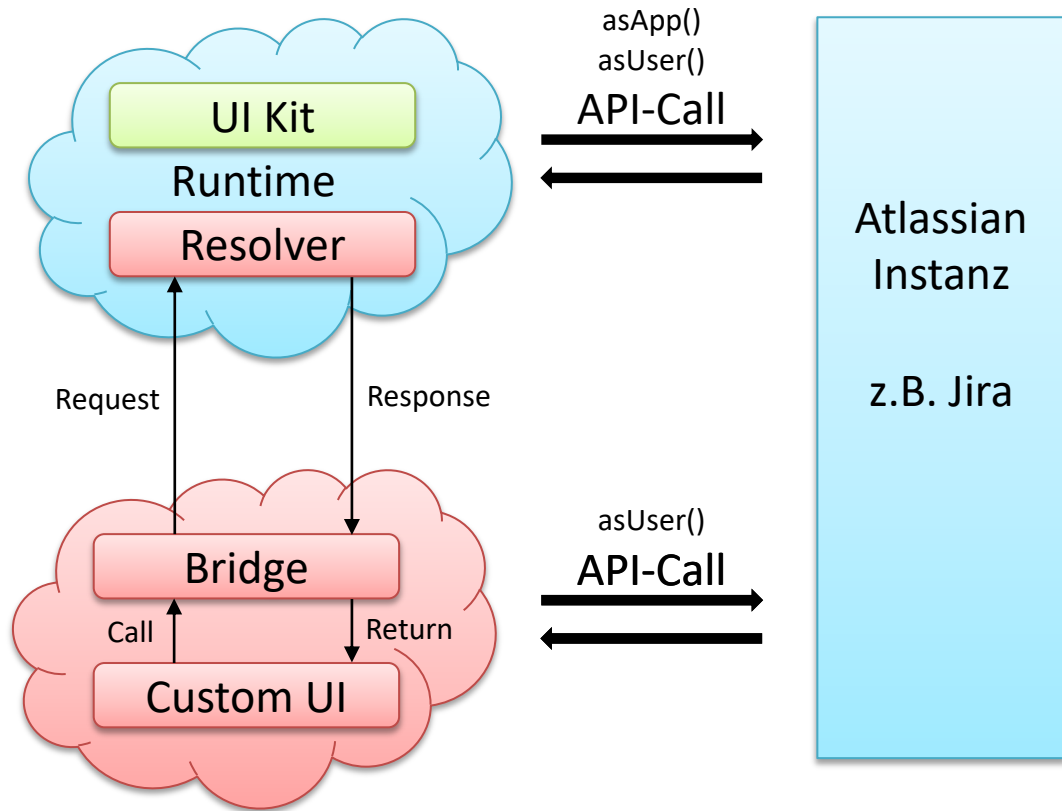
Complexity: LOW

Test Intensity: 13

Values are computed

App installieren

©Engin_Akyurt/pixabay.com



Die Forge Architektur



2. Die Forge-Architektur ist komplex



- Aufteilung in Runtime und Custom UI hilft bei der Planung
- Routine macht auch Komplexes alltäglich

Custom UI Resolver

- Definition

Muss mit npm/yarn
werden!

**API Limits
beachten
!**

Fetch API

- Basic
- Product Aut

Storage API

- Key –Value Sto
- Per Installation

Forge Runtime und API

```
.../resolver';
```

```
...();
```

```
=> {
```

```
...getDefinitions();
```

```
App()
```

```
User()
```

API-Call

```
'@forge/api';
```

```
"https://example.net/";
```

```
...a(route('/rest/api/3/field'));
```

```
scopes:
```

```
3
```

```
- storage:app
```

- Für „einfache“ App-Teile
- React-ähnlicher Syntax
- Produkt-spezifisch

Einschränkungen

- Kein aktualisieren der View
- Forge Komponenten \neq React Komponenten
- Schwer mit Unittests abdeckbar
- Nur eingeschränktes Set an UI-Elementen

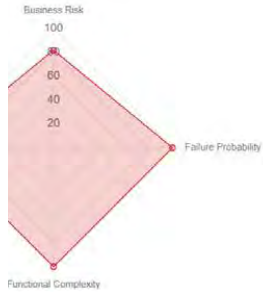
UI Kit Hooks

- React Hooks

```
1 import ForgeUI, { useProductContext } from "@forge/ui";  
2  
3 const platformContext = useProductContext().platformContext
```

Forge UI Kit

Journey To Rome



To Do

Actions

Details

Assignee

Unassigned

Assign to me

Reporter

Daniel Kleissl

Labels

None

Risk evaluation DEV

Risk: A1

Complexity: HIGH

Test Intensity: 100

Values are computed

Priority

Medium

More fields Original esti...

Custom Field

Risk Assessment

Use values from wizard

Edit values

Risk

A1

Complexity

High

Test Intensity (1-100)

100

Cancel

Save

Custom Field Edit

Forge UI Kit

Debugging via Tunnel

```
1 forge tunnel
```

```
Listening for requests...  
Serving file index.html for resource wizard  
invocation: 9c64e60d85df41be risk  
INFO 07:56:19.466 9c64e60d85  
invocation: 983d425839303120 i  
invocation: a32e4791628665ec index
```

Auch wenn App aktualisiert
werden soll!

	Name	Image	Status
<input type="checkbox"/>	forge-tunnel-docker	atlassian/forge-tunnel:latest	Running



Größere Änderungen im Manifest

- Berechtigungen
- Module

```
2 forge deploy  
3 forge install --upgrade
```

Development Workflow



3. Einfache Projekte? Verwende das UI Kit

- Ermöglicht dir einfache Apps zu erstellen
- Für komplexere Projekte ist ein Custom UI zu empfehlen
- Auch für erfahrenere Developer nützlich!

One
more
thing



Statische Resource via Iframe


Default Content Security Policy:

Externe Scripts/Assets

- Externe Bilder
- Externe Scripts

Inline CSS

Einschränkungen

- Forge-API nur über Bridge-API
- „Hot Reload“ → Änderung im Manifest
- Kommunikation UI Kit  Custom UI

Custom Content Security Policy:

Externe Assets via Bridge

```
1 'script-src https://www.example.com'
2
3
4
5
6
7 content:
8   styles:
9     - unsafe-inline
```

Forge Custom UI

Muss mit npm
installiert

API Limits beachten !

Muss mit npm
installiert

```
1 import { invoke } from '@forge/bridge'
2
3 const getFoo = async () => await invoke('foo')
```

```
1 import { view } from '@forge/bridge'
2
3 await view.refresh();
4 const context = await view.getContext()
```

nsApp()
sUser()
I-Call

```
requestJira, requestConfluence } from '@forge/bridge';
response = await requestJira('/rest/api/3/issue/ISSUE-1');
confluenceResponse = await requestJira('/wiki/rest/api/content');
```

()
Call

Custom UI Bridge API

Invocations

- Anzahl
- Laufzeit der Anfrage
- Returnierte Datenmenge

Ressource Updates

- Für Custom UI

Storage

- Kapazität
- Datenmenge Lesen/Schreiben

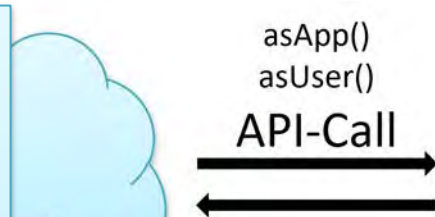
Memory

- App zu Laufzeit

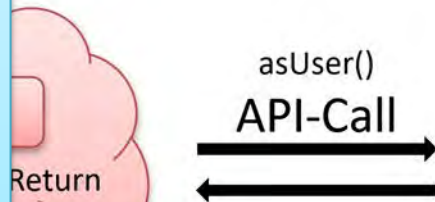
Size

- App Gesamtgröße

Async Events (API)



Response



Return

Forge API Limits

**Nicht abschrecken
lassen!**



Calculate FaaS usage

The Jira issue activity panel loads when the resolver takes an average of 500 ms to resolve an issue in Jira.

In this example, the view issue event consumes one FaaS invocation, 500 ms of execution time, and 32 KB of memory per seat for the 500-seat licenses and 125,000 issue views.

- 1 $(78 * 50) + (25 * 5000)$
- 2 $(128,500 * 500 \text{ ms}) / 1000$
- 3 $(128,500 * 32 \text{ KB}) / 1000$

Forge API Limits

View and filter app metrics to see how your app is performing. To learn more about each metric, see Monitor your apps.

All sites

Development

2023-03-30 10:00

Reset

From Mar 30 06:00 (UTC) to Apr 12 06:00 (UTC) | Last updated a few seconds ago

Invocation success rate

100%

Overall

Invocation success rate (5)

Percentage of successful invocations

Group by: None

%

100

75

50

25

0

Invocation time

Time ranges showing speed of function invocations

Group by: None

%

100

75

50

25

0

0

25

50

75

100

125

150

175

200

225

250

275

300

325

350

375

400

425

450

475

500

525

550

575

600

625

650

675

700

725

750

775

800

825

850

875

900

925

950

975

1000

1025

1050

1075

1100

1125

1150

1175

1200

1225

1250

1275

1300

1325

1350

1375

1400

1425

1450

1475

1500

1525

1550

1575

1600

1625

1650

1675

1700

1725

1750

1775

1800

1825

1850

1875

1900

1925

1950

1975

2000

2025

2050

2075

2100

2125

2150

2175

2200

2225

2250

2275

2300

2325

2350

2375

2400

2425

2450

2475

2500

2525

2550

2575

2600

2625

2650

2675

2700

2725

2750

2775

2800

2825

2850

2875

2900

2925

2950

2975

3000

3025

3050

3075

3100

3125

3150

3175

3200

3225

3250

3275

3300

3325

3350

3375

3400

3425

3450

3475

3500

3525

3550

3575

3600

3625

3650

3675

3700

3725

3750

3775

3800

3825

3850

3875

3900

3925

3950

3975

4000

4025

4050

4075

4100

4125

4150

4175

4200

4225

4250

4275

4300

4325

4350

4375

4400

4425

4450

4475

4500

4525

4550

4575

4600

4625

4650

4675

4700

4725

4750

4775

4800

4825

4850

4875

4900

4925

4950

4975

5000

5025

5050

5075

5100

5125

5150

5175

5200

5225

5250

5275

5300

5325

5350

5375

5400

5425

5450

5475

5500

5525

5550

5575

5600

5625

5650

5675

5700

5725

5750

5775

5800

5825

5850

5875

5900

5925

5950

5975

6000

6025

6050

6075

6100

6125

6150

6175

6200

6225

6250

```

1 modules:
2   jira:issuePanel:
3     - key: example-app
4       resource: wizard
5       resolver:
6         function: resolver
7       title: Example App
8       viewportSize: large
9       tooltip: An app example

```

asApp()

```

1 modules:
2   function:
3     - key: resolver
4       handler: index.handler

```

Runtime
Resolver

Request

Response

```

1 import Resolver from '@forge/resolver';
2
3 const resolver = new Resolver();
4
5 resolver.define('get-foo', () => {
6   return 'bar';
7 });
8
9 export const handler = resolver.getDefinitions()

```

asUser()

API-Call

```

1 resources:
2   - key: wizard
3     path: static/wizard/build

```

Bridge

Call

Return

Custom UI

Manifest Module

Jira Software Your work ▾ Projects ▾ Filters ▾ Dashboards ▾ Teams ▾ Apps ▾ Create

test2 Software project

PLANNING

TES Board Board

Roadmap

Backlog

Active sprints

Reports

Issues

Components

DEVELOPMENT

Code

Releases

Project pages

Add shortcut

Projects / test2 / TES-4

test task

Attach Create subtask Link issue Journey To Rome ...

Description

Add a description...

Issue Panel

Journey To Rome DEV

Process

- Only UI
- Calculation/Validation
- Data Change

What is affected by the required process/functionality?

- ☐ Only UI ⓘ
- ☐ Calculation Validation ⓘ
- ☐ Data change ⓘ

1 2 3 4 5 6 7 8 9 10 11 12 Abort Next question

Assignee

Unassigned

Assign to me

Reporter

Daniel Kleissl

Labels

None

Risk evaluation DEV

Risk: A1

Complexity: HIGH

Test Intensity: 100

Values are computed

Priority

Medium

More fields Original estimat...

„Hot Reload“ → Eintrag im Manifest

- via React-Dev-Server

Runtime

Resolver

```
1 forge deploy
2
3 forge install --upgrade
```

Request

Response

```
1 resources:
2   - key: wizard
3     path: static/wizard/build
4     tunnel:
5       port: 3000
```

```
1 forge tunnel
```

App aktualisieren

1. Build von Custom UI
2. Deploy via CLI

Bridge

Call

Custom UI

```
1 react-scripts start
```

```
1 react-scripts build
```

Development Workflow



4. Komplexe Projekte? Verwende das Custom UI

One
more
thing

- Du kannst alle Node Module nutzen, die React-kompatibel sind
- Fast alle Module unterstützen das Custom UI

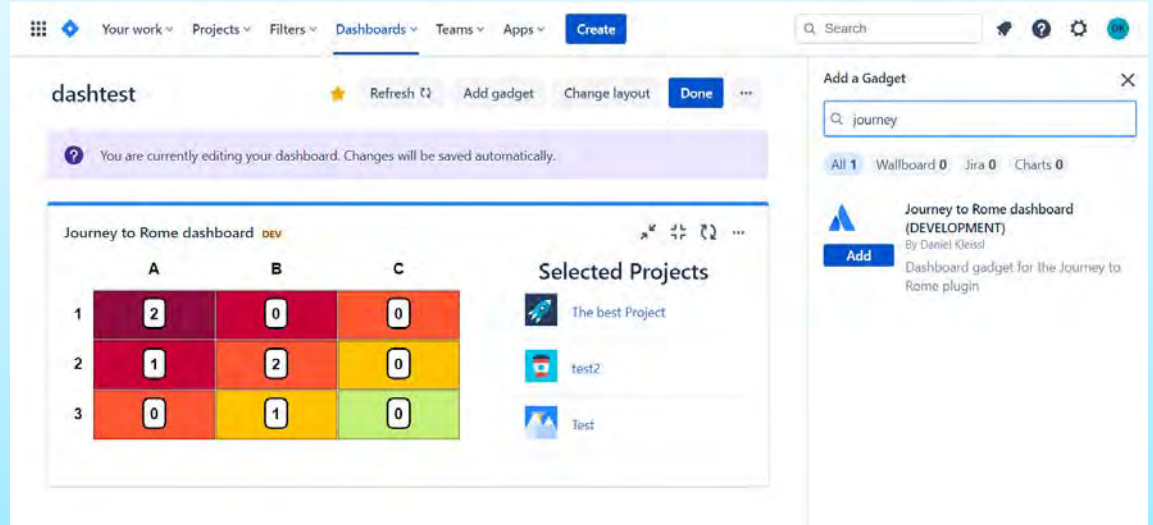
Manifest

- Berechtigungen
- Ressourcen

Module

- Wo wird App dargestellt
- Zusätzliche Funktionen
- App-Informationen
- Icons

Dashboard Modul

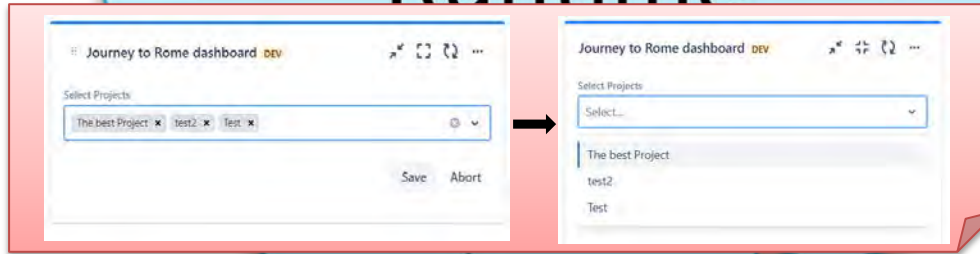


Manifest und Module

Relativ neu Bundle App+Dashboard

```
1 modules:
2   jira:dashboardGadget:
3     - key: example-dashboard-gadget
4       title: Example Dashboard
5       description: Example Dashboard
6       thumbnail: https://developer.atlassian.com/platform/forge/images/icons/issue-panel-icon.svg
7       resource: dashboard
8       resolver:
9         function: resolver
10      edit:
11        resource: dashboard
12    function:
13      - key: resolver
14        handler: index.handler
15  resources:
16    - key: dashboard
17      path: static/dashboard/build
```

Bug in Iframe



Workaround

```
1 document.documentElement.style.overflowY = 'visible';
```

Dashboard Modul

Abhängig von Atlassian Produkt
App-Berechtigungen im Manifest

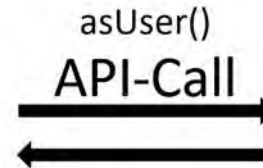
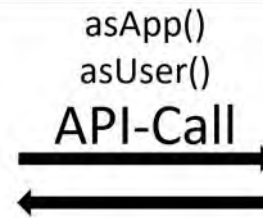
Für Development

- „Classic“ Scopes verwenden
- Einschränken als letzter Schritt

Nach Installation muss Berechtigungen
zugestimmt werden

```
1 permissions:
2   scopes:
3     - read:field:jira
4     - write:issue:jira
5     - read:jira-user
6     - read:jira-work
```

Atlassian Produkt API Berechtigungen



Atlassian
Instanz

z.B. Jira



5. Beachte das Principle of Least Privilege

- Versuche, so wenig Berechtigungen wie möglich vom User anzufordern
- Keine Scheu vor Reverse Engineering bei Modulen

One
more
thing

UI KIT

Layout  Tests

Logic Tests nur von Forge-Modulen getrennt

Kit



runtime

Resolver

Muss vor jedem anderen File,
das @forge/bridge importiert,
gemocked werden!

Custom UI

Voraussetzung

- „React Testing Library“ (o.ä.) 
- Mock-Framework z.B. Sinon.js 

Vorbereitung

- Mock der Forge Bridge

Testen analog zu React

```
1 import sinon from 'sinon';
2
3 window.__bridge = {
4   callBridge: sinon.stub()
5 };
```

idge

App testen

Custom UI



6. Testbarkeit wichtig? Verwende das Custom UI

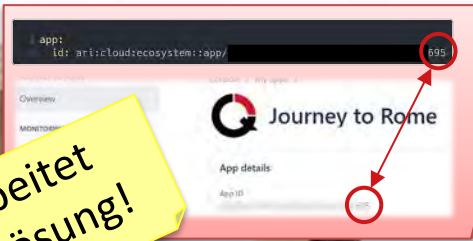
One
more
thing

- UI Kit Layouts sind derzeit nicht testbar
- Nicht jeder Error ist auch euer Fehler



Herausforderung

Verknüpfung AppID und Dev-Account



Atlassian arbeitet
an nativer Lösung!

Unsere Lösung

Ein Account pro Developer

- Atlassian Instanz
- App Registration

Ein „Master“-Account

- Produktiv-Umgebung
- Aktueller Release-Stand

AppID switching via Script

1. AppID update

- Setzt Developer AppID

2. Build und Deploy

- Deploy auf Environment

3. AppID reset

- Reset auf Master AppID

Developer Collaboration

©Andrii Yalanskyi/Shutterstock.com



7. Verwende mehrere Instanzen zur Kollaboration



- Ermöglicht deinem Team, unabhängig Features umzusetzen
- Definiere eine Instanz als Produktivumgebung um den Release Stand zu tracken

Herausforderung

- Installation ID nicht isoliert abfragbar
- *forge install* bei vorhandener Installation
- *forge uninstall* wenn keine Installation

```
PS C:\Users\daniel.kleissl\Documents\razzfazz\repos\RnR\plugin> forge install list
```

Showing all the current installations of your app:

Installation ID	Environment	Site
	development	danielkleissl.atlassian.net

Unsere Lösung

1. Abfrage Installation ID
 - Via *grep* und *regex*
2. Deinstallation mit ID (wenn installiert)
 - *forge uninstall [Installation Id]*
3. Installation von neuer Version
 - *forge install*

Automation mit CI/CD

©Andrii Yalanskyi/Shutterstock.com



8. Forge App Deployment per CI/CD ist schwer, aber machbar

- Erleichtert verschiedene Umgebungen auf aktuellen Stand zu halten
- Behalte die Forge Roadmap im Auge

One
more
thing

Voraussetzung

- App sharing aktiv
- Auf Production deployed
- Vendor Account
- Developer Community Account

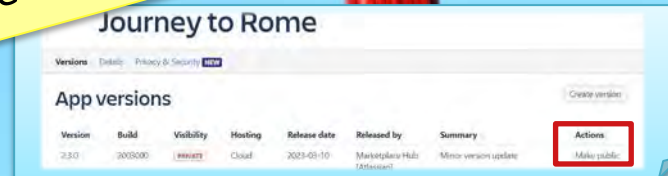
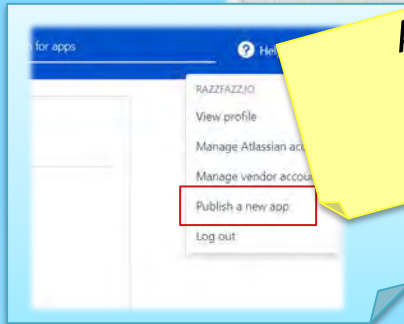
Optional

- Security Self-Assessment
- Support Portal Link

Vor Release beantworten

- Gratis oder bezahlte Lizenz?
- App approval guidelines erfüllt?
- Welches Support Service Level Agreement?
- Welches end user license agreement (EULA)?
- Datenschutz-Grundverordnung eingehalten?
- Dokumentation vollständig *und* zugänglich?

App released →
sharing via
Developer Console



Release im Marketplace



9. Plane den Release im Marketplace gut

- Erstelle alle nötigen Marketing-Assets und Screenshots vor dem Release
- Du müsst die App nicht releasen um sie zu teilen

One
more
thing

▲ **ATLASSIAN** Developer Community

Interaktion mit Atlassian Experten
Einmelden von Bugs

▲ **ATLASSIAN** Developer

Blog
Aktuelle Issues
Changelog

▲ **ATLASSIAN** Marketplace

App Beispiele
EULA Beispiele von anderen Apps

■ **Trello**

Roadmap
Zukünftige Features

Zusätzliche Ressourcen

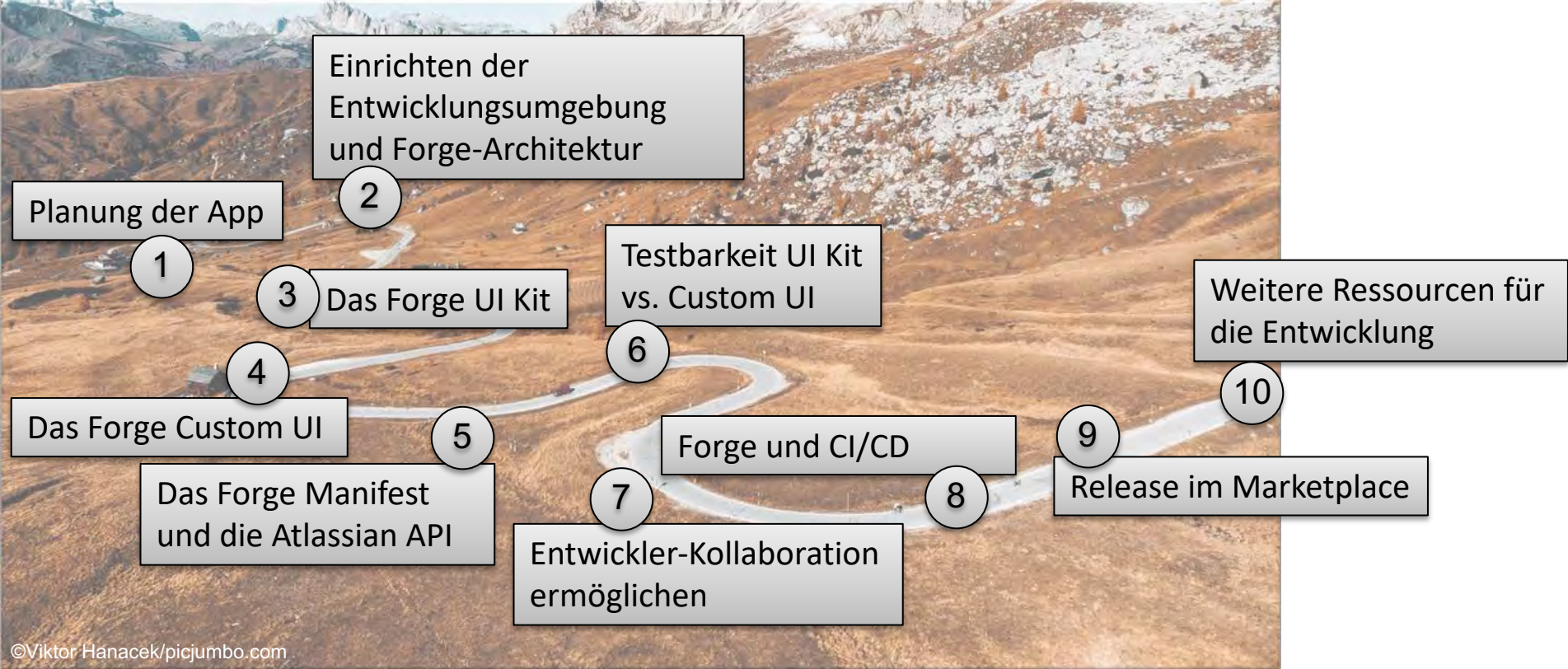
©Viktor Hanacek/picjumbo.com



10. Nutzt die Stärke der Community

- Die Atlassian Developer Community ist der beste Ort, um Hilfe bei Herausforderungen zu bekommen
- Bugs einmelden hilft!

One
more
thing



1. Plane den Aufstieg deiner App in die Cloud
2. Die Forge-Architektur ist komplex
3. Einfache Projekte? Verwende das UI Kit
4. Komplexe Projekte? Verwende das Custom UI
5. Beachte das Principle of Least Privilege
6. Testbarkeit wichtig? Verwende das Custom UI
7. Verwende mehrere Instanzen zur Kollaboration
8. Forge App Deployment per CI/CD ist schwer, aber machbar
9. Plane den Release im Marketplace gut
10. Nutzt die Stärke der Community