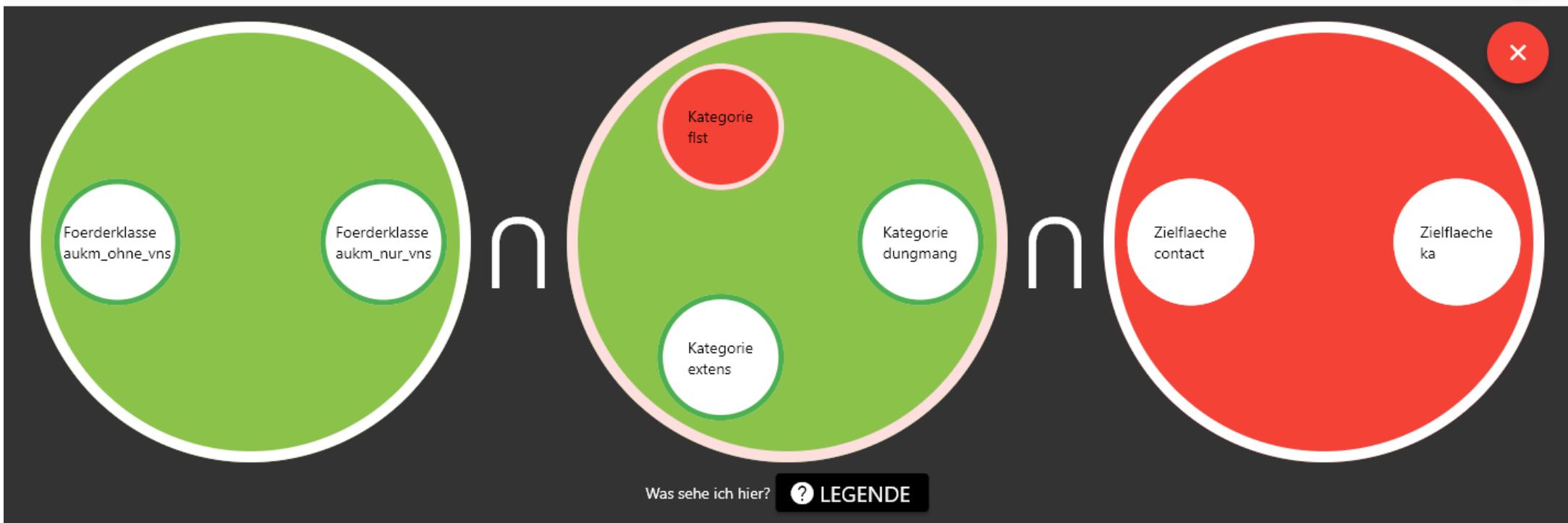


[← Zieleinheit](#)

- keine Angabe/Vorgabe (i)
- ha (i)
- bitte um Unterstützung (i)
- m³ (z.B. Gülle) (✓)





Entwicklung einer Formularanwendung
mit Kompatibilitätsvalidierung der
Einfach- und Mehrfachauswahl-Eingabefelder

← Maßnahmen Detail DEBUG

Status
in Bearbeitung 

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz 

Kategorie 

Zielsetzung

Zielfläche 

Zieleinheit 

Hauptzielsetzung Land 

Nebenziele 

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

Maßnahmen Detail

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

Kategorie

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- m³ (z.B. Gülle)
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung

✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

keine Angabe/Vorgabe

m³ (z.B. Gülle)

Kopf/Stück (z.B. Tiere oder Bäume)

GV/GVE

RGV

ha

bitte um Unterstützung

✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit
 m³ (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- m³ (z.B. Gülle)
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung

✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit
 m³ (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- m³ (z.B. Gülle)
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung

✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit
 m^3 (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zieleinheit
 m^3 (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

Zielsetzung

Zielfläche

Zieleinheit
 m^3 (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

Zielsetzung

Zielfläche

Zieleinheit
 m^3 (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

- ← Kategorie DEBUG
- Anbau Zwischenfrucht/Untersaat
 - Anlage/Pflege Struktur
 - Düngemanagement
 - Extensivierung
 - Flächenstilllegung/Brache
 - Nutzungsumwandlung
 - Förderung bestimmter Rassen / Sorten / Kulturen
 - bitte um Unterstützung
- ✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

Zielsetzung

Zielfläche

Zieleinheit
m³ (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

keine Angabe/Vorgabe

m³ (z.B. Gülle)

Kopf/Stück (z.B. Tiere oder Bäume)

GV/GVE

RGV

ha

bitte um Unterstützung

✓

Maßnahmen Detail

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

Zielsetzung

Zielfläche

Zieleinheit
 m³ (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele

Zieleinheit

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)

✓

← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

Zielsetzung

Zielfläche

Zieleinheit

Hauptzielsetzung Land

Nebenziele

← Zieleinheit DEBUG

keine Angabe/Vorgabe

Kopf/Stück (z.B. Tiere oder Bäume)

GV/GVE

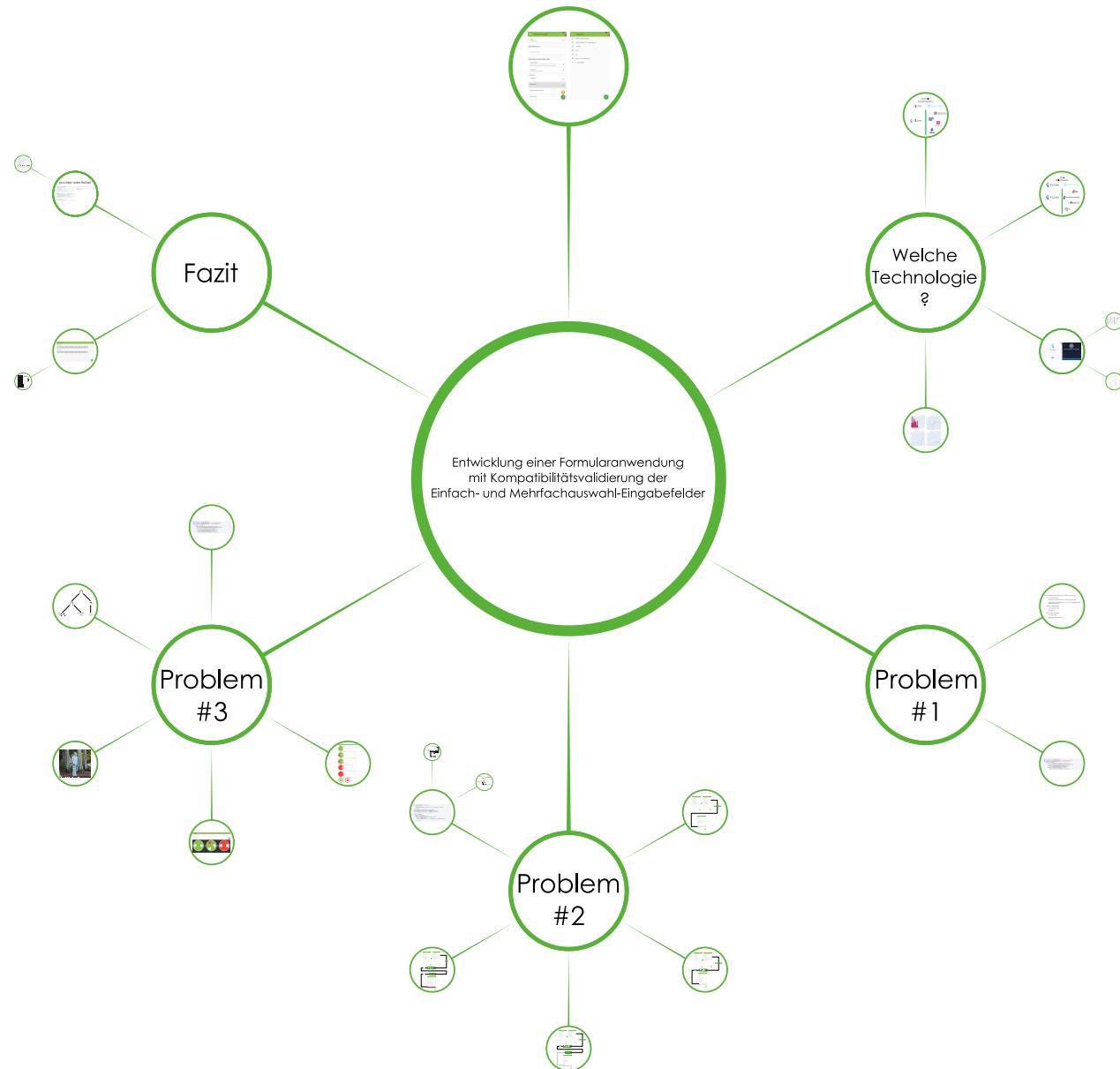
RGV

ha

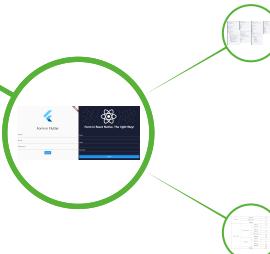
bitte um Unterstützung

✗ m³ (z.B. Gülle)

✓



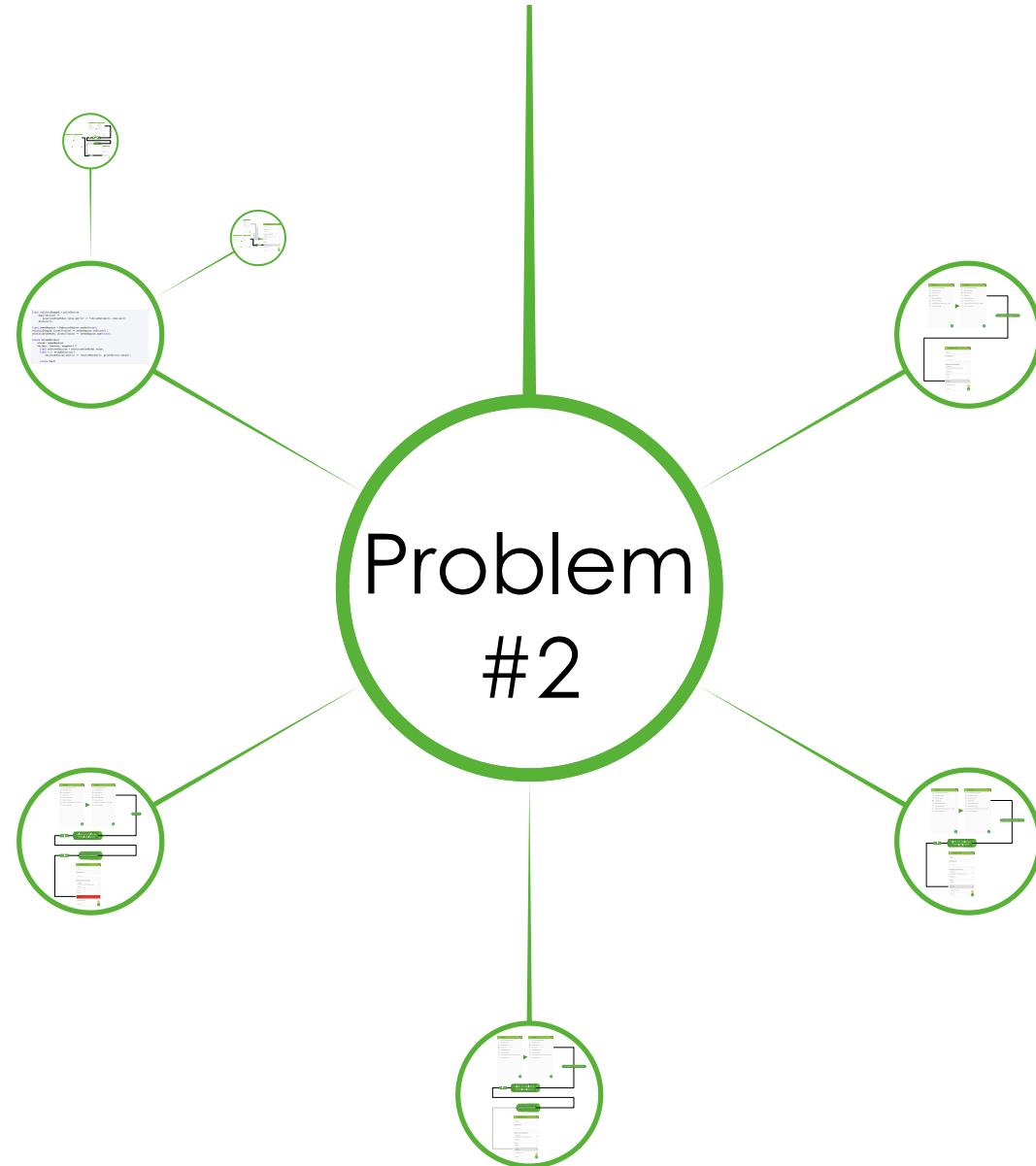
Welche Technologie ?



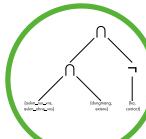
Problem #1

m





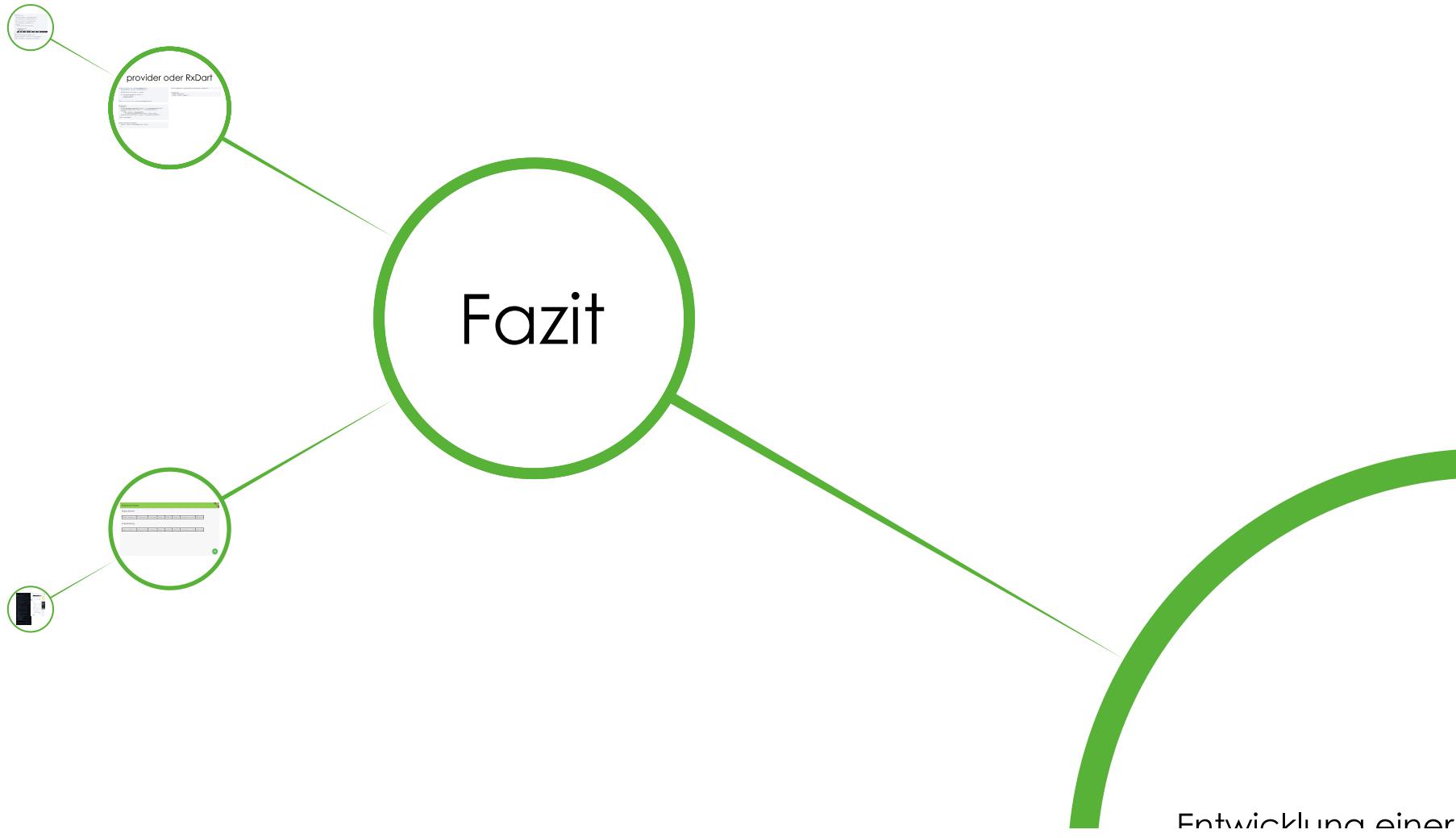
Problem #3



Über Konzepte der Datenbank
Unter einer **Relation** versteht man eine Tabelle mit
Attributnamen (Spalten) und Attributwerten (Werte).
Attributnamen sind **Attributbeschreibung**, Attributwerte
sind **Attributrealisierung**.
Attributbeschreibungen sind **Attributnamen**, Attributrealisierungen
sind **Attributwerte**.
Attributbeschreibungen sind **Attributnamen**, Attributrealisierungen
sind **Attributwerte**.



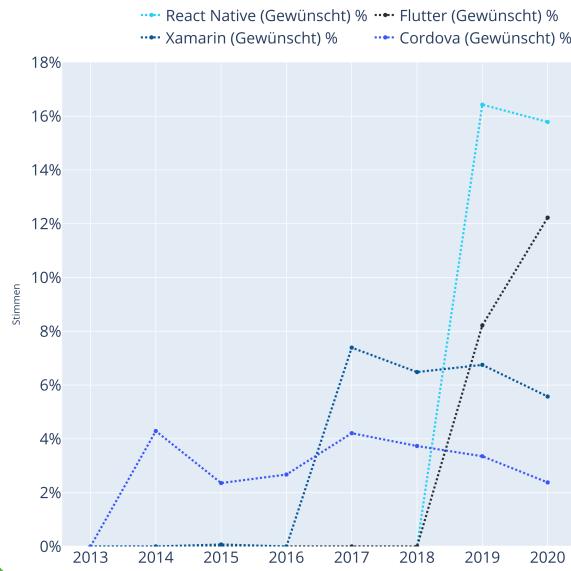
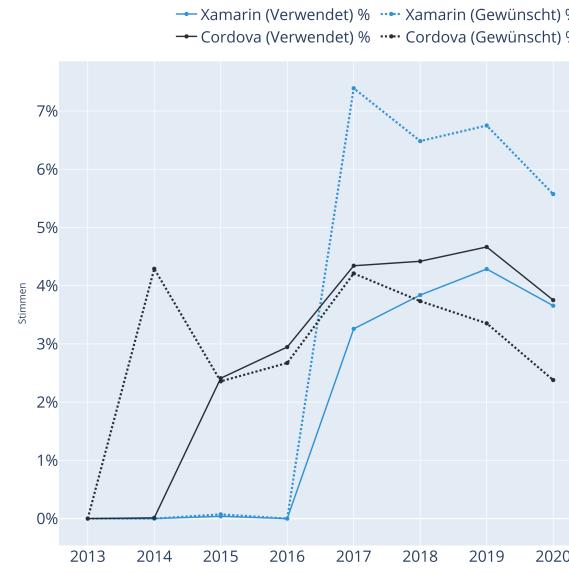
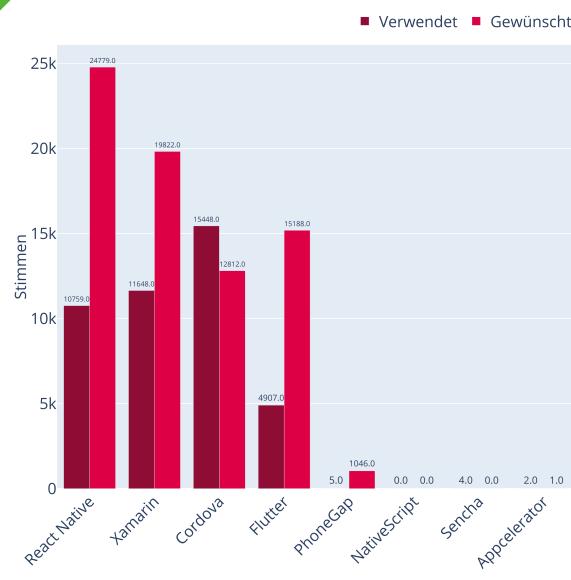
© 2019-2020 MERRILL

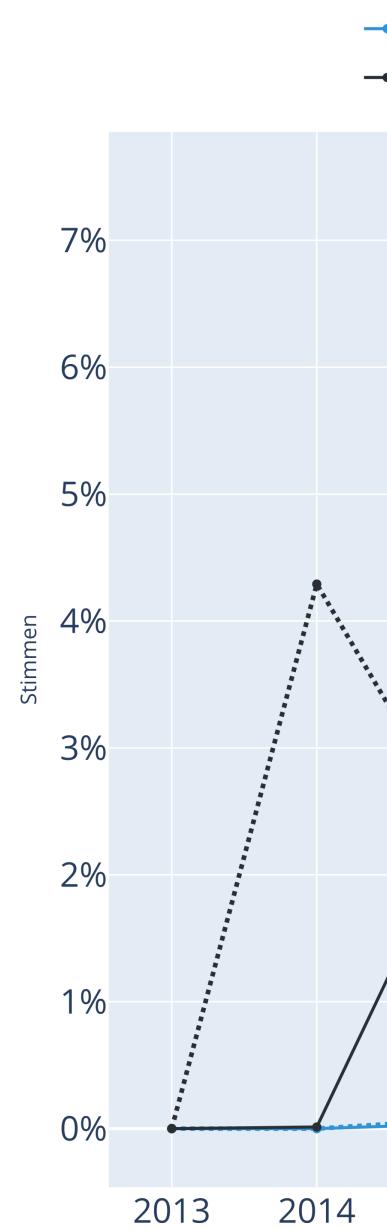
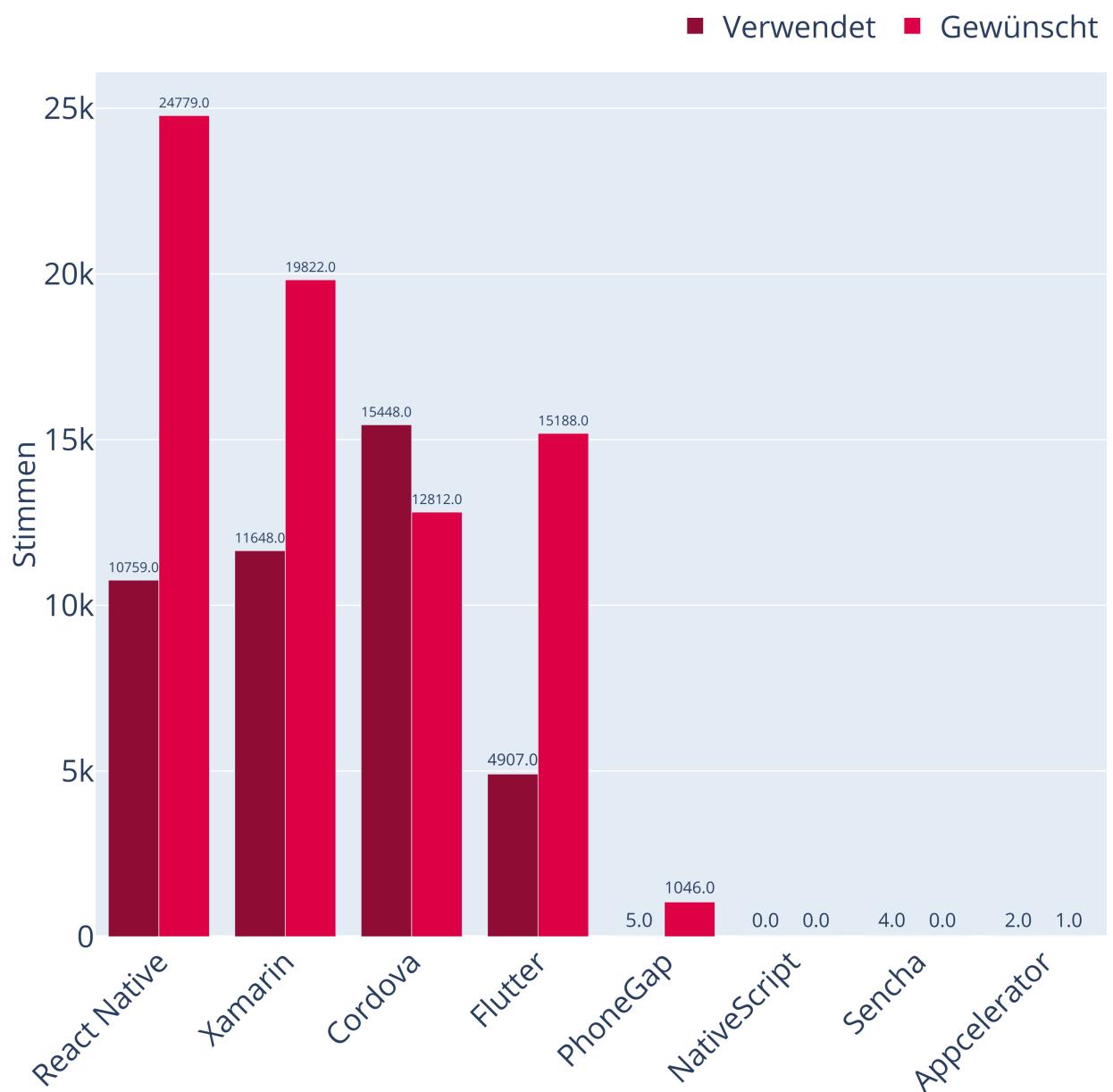


Welche Technologie ?

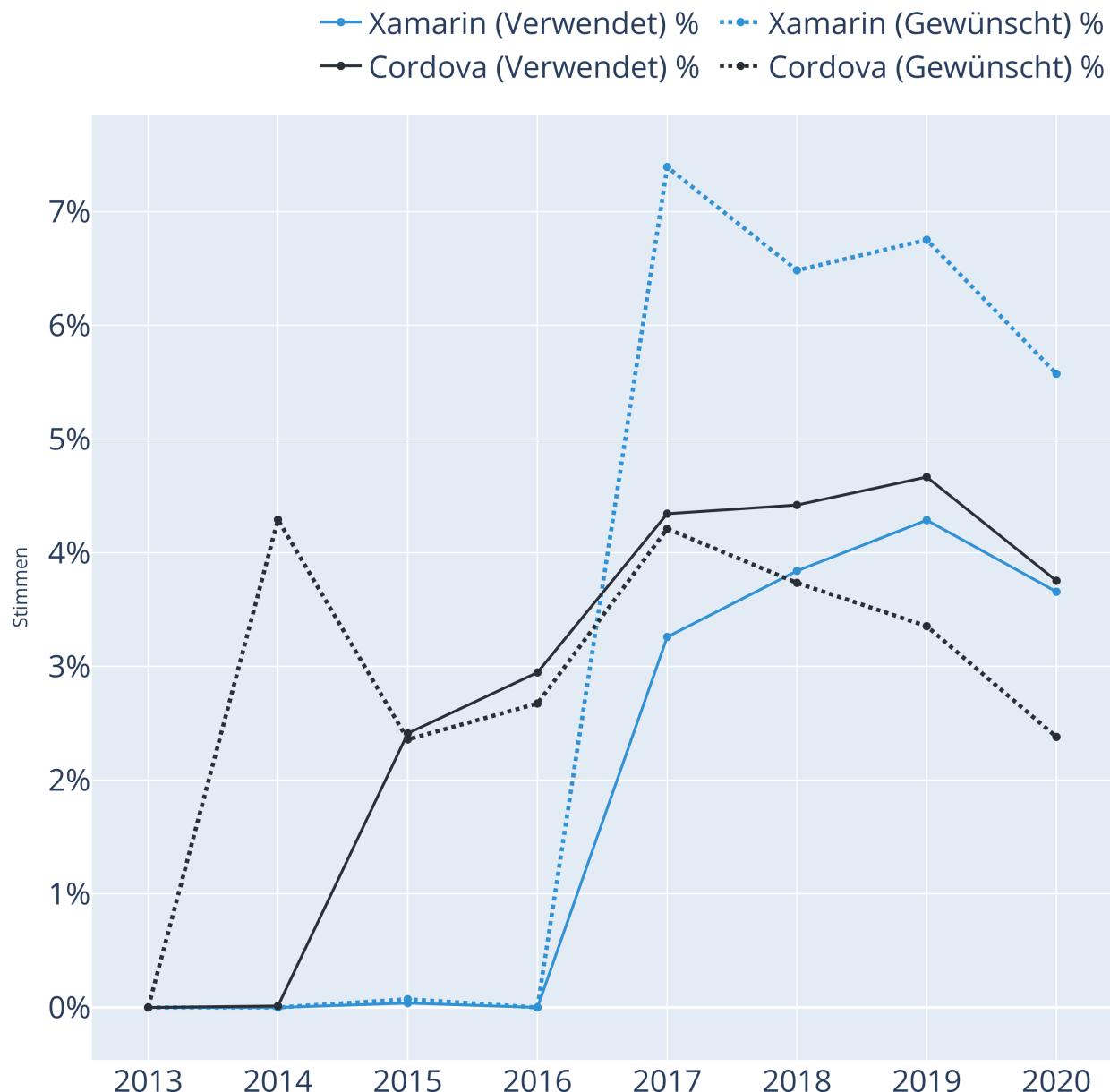
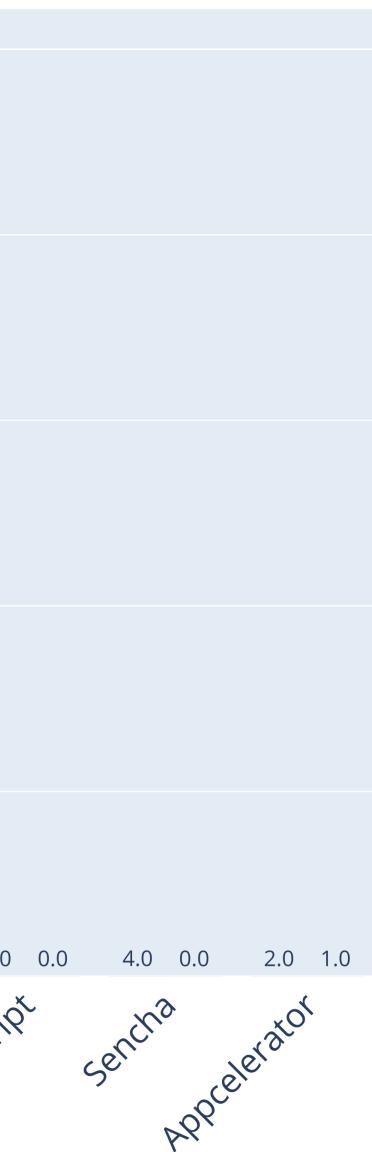


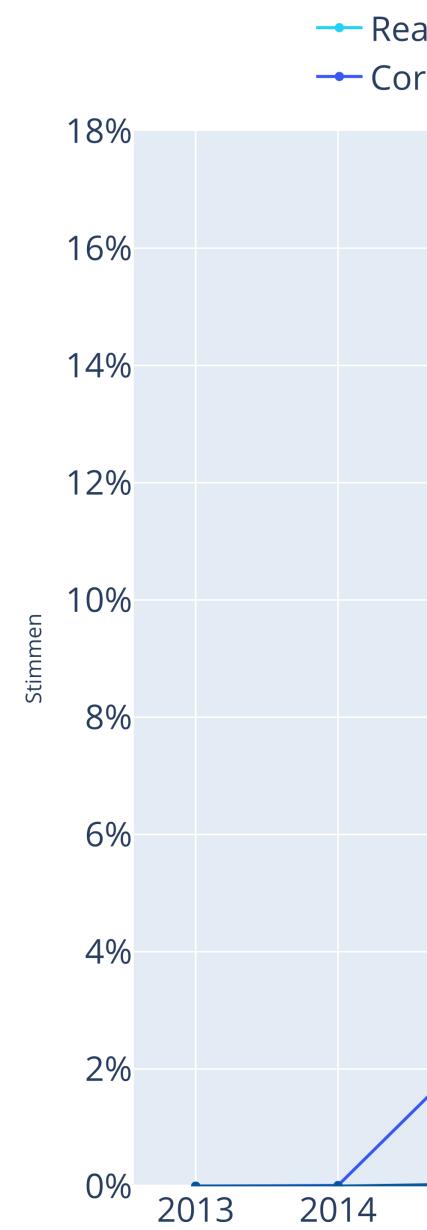
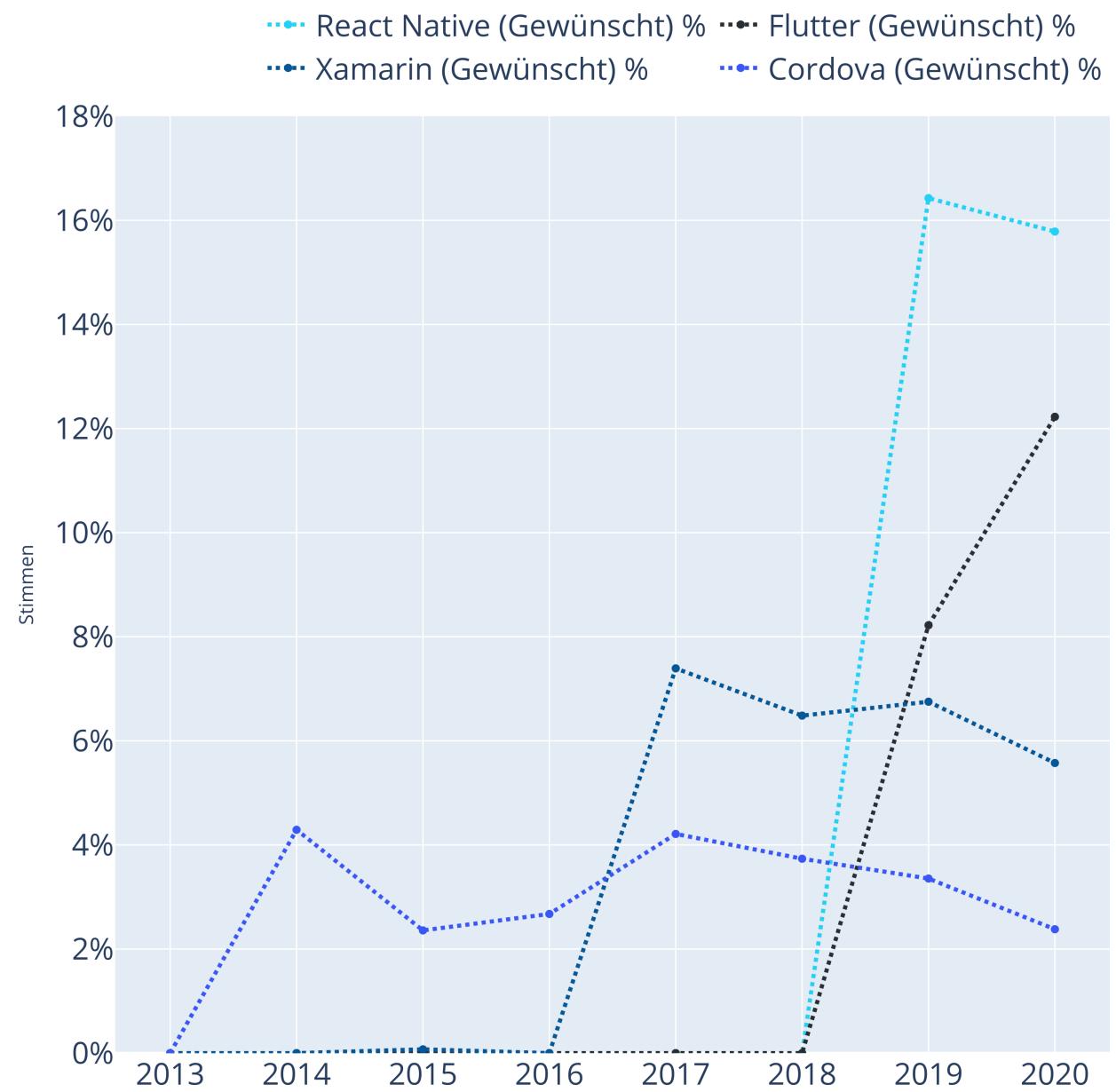
Anwendung



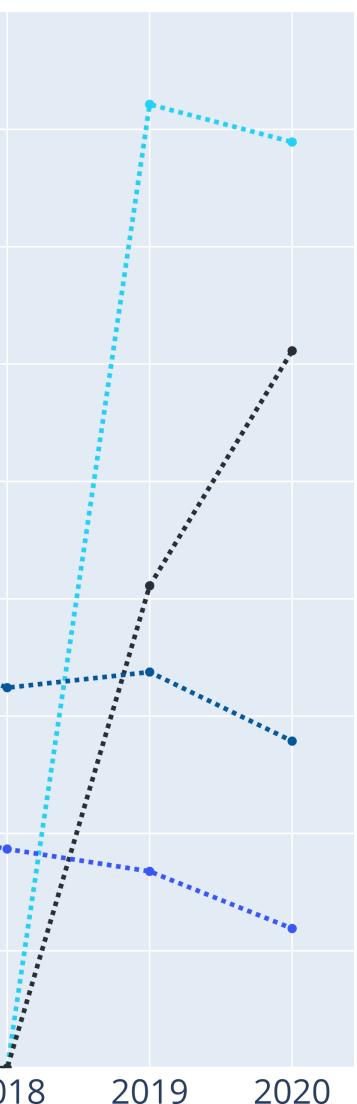


wendet ■ Gewünscht

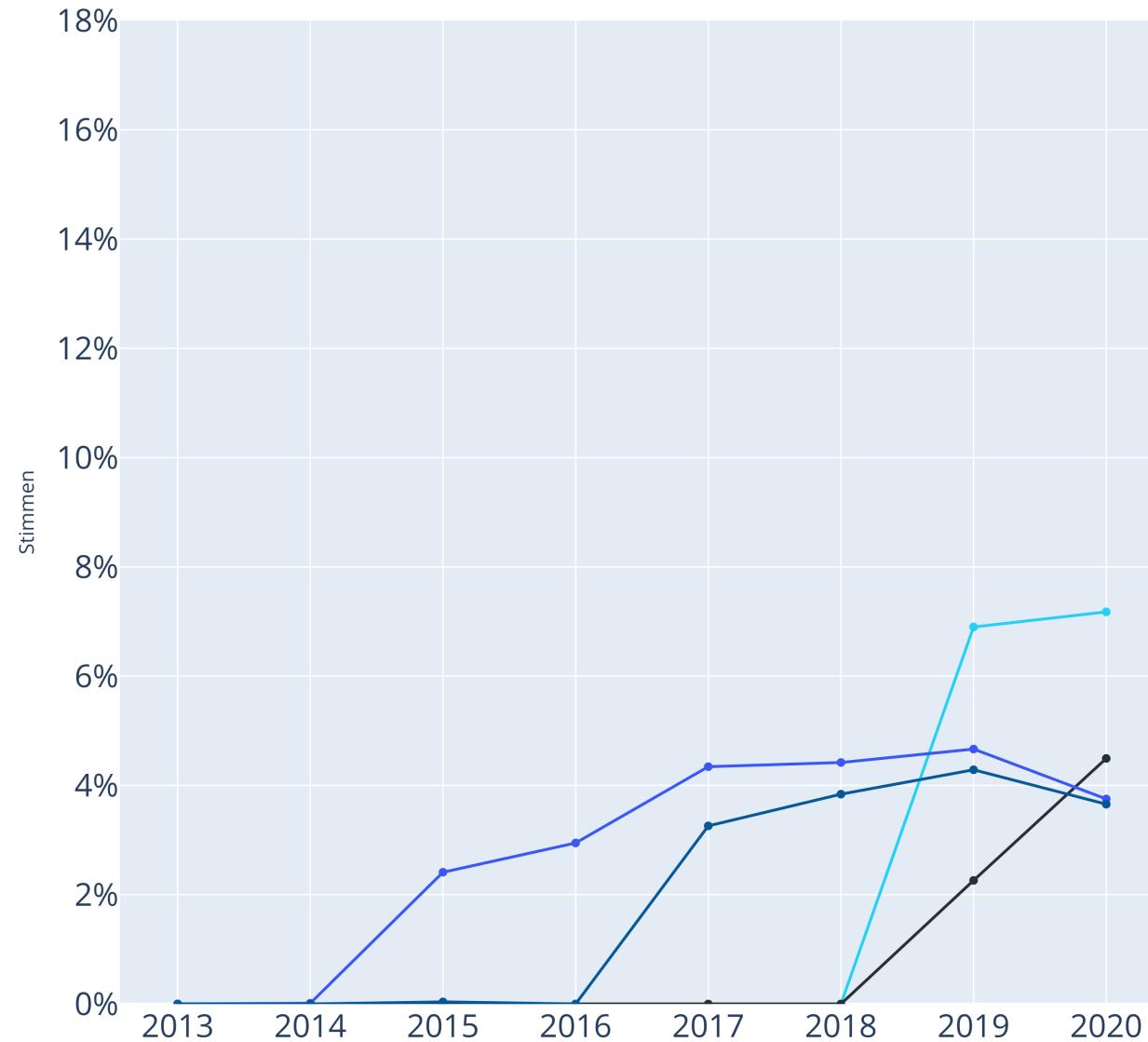




Flutter (Gewünscht) %
Cordova (Gewünscht) %

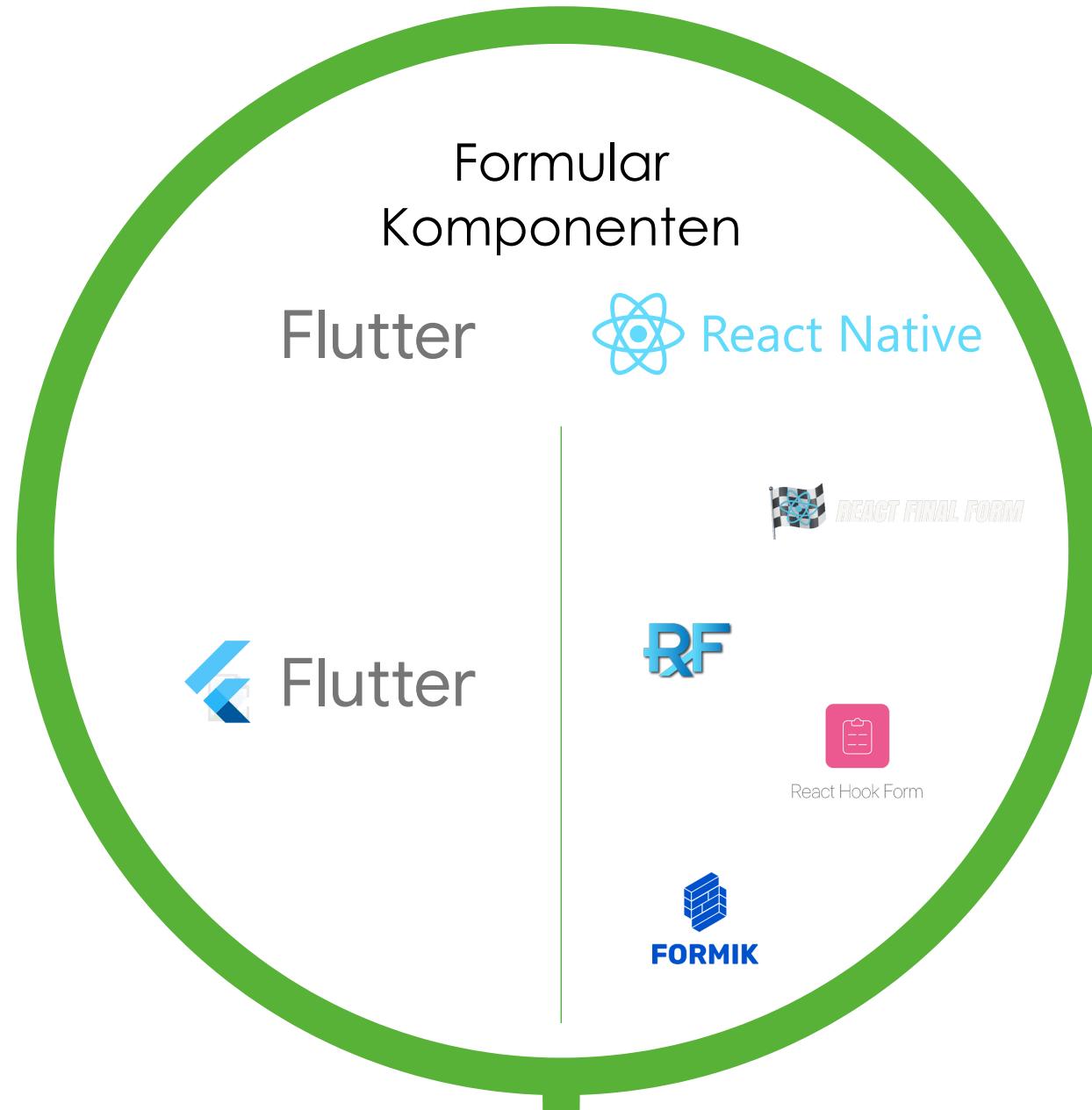


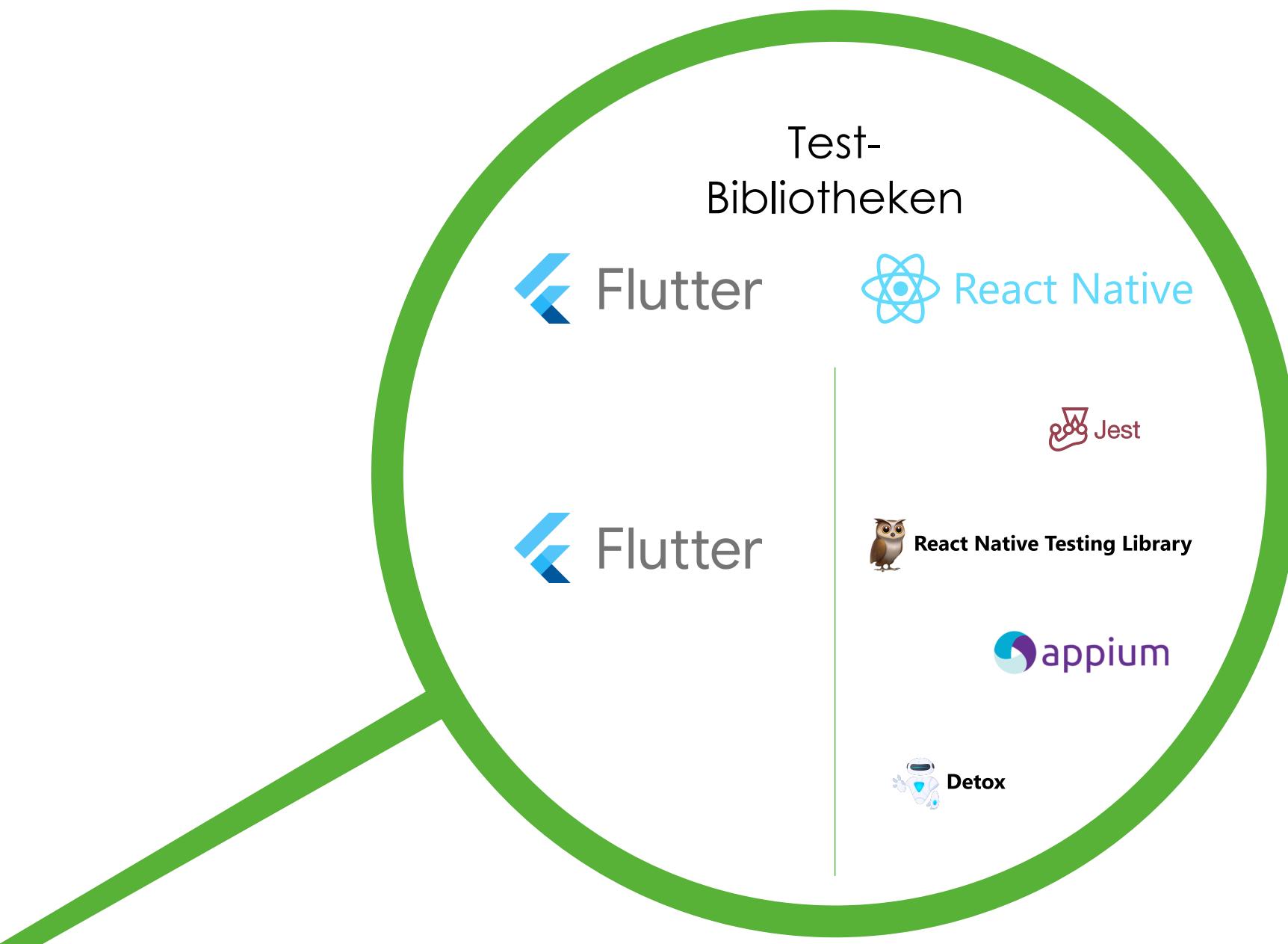
React Native (Verwendet) % Flutter (Verwendet) %
Cordova (Verwendet) % Xamarin (Verwendet) %

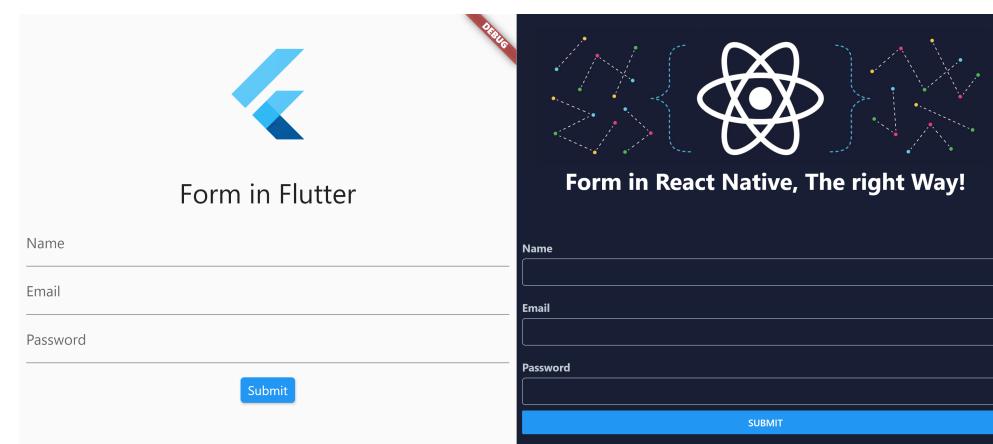


Welche Technologie ?

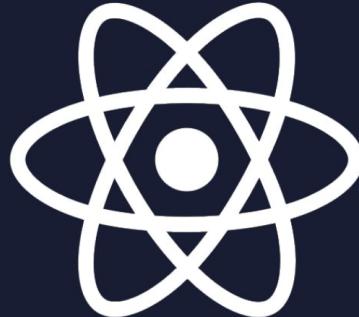








DEBUG



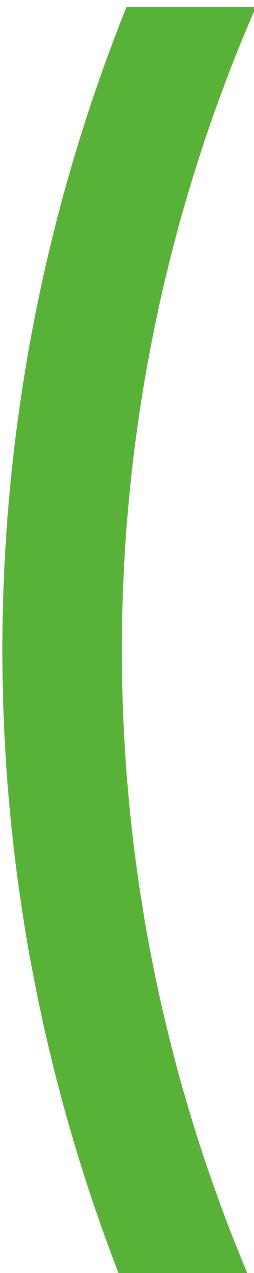
Form in React Native, The right Way!

Name

Email

Password

SUBMIT



DEBUG

Form in Flutter

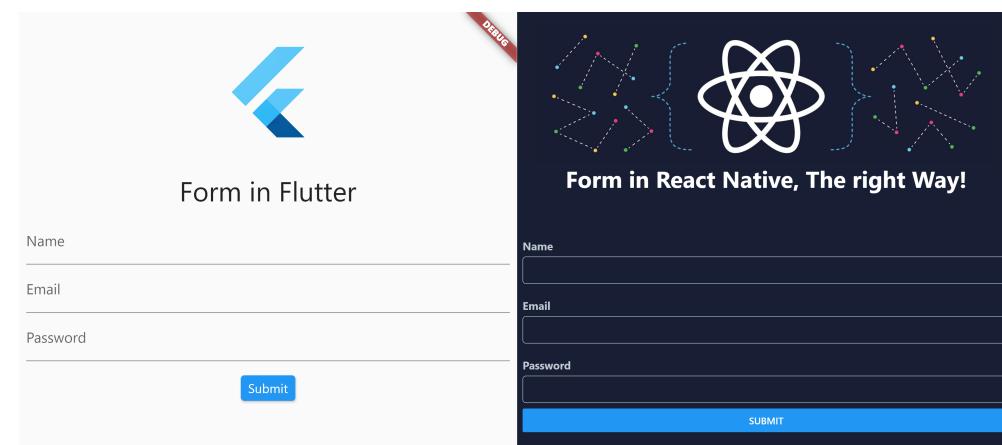
Name

Email

Password

Submit






```
1 final emailPattern = RegExp(  
2     r'^(([^<>()\\[\\]\\\\.,:\\s@"]+(\.([^<>()\\[\\]\\\\.,:\\s@"]+)*|(.+))@((\\[[0-9]{1,3}\\._  
3         [0-9]{1,3}\\_.[0-9]{1,3}\\_.[0-9]{1,3}\\])|(([a-zA-Z\\-0-9]+\\.)+[a-zA-Z]{2,}))$');  
4  
5 String? validateEmail(String label, String? value) {  
6     if (value == null || value.isEmpty) {  
7         return '$label is required';  
8     } else if (!emailPattern.hasMatch(value)) {  
9         return 'Invalid Email Format';  
10    } else {  
11        return null;  
12    }  
13}  
14  
15 String? validateNotEmpty(String label, String? value) {  
16     if (value == null || value.isEmpty) {  
17         return '$label is required';  
18     } else {  
19         return null;  
20     }  
21}  
22  
23 void main() => runApp(MyApp());  
24
```

```
44         onChanged: (value) => formData["email"] = value),
45     TextFormField(
46         decoration: const InputDecoration(labelText: "Password"),
47         validator: (String? value) =>
48             validateNotEmpty("Password", value),
49         onChanged: (value) => formData["password"] = value),
50     Padding(
51         padding: const EdgeInsets.symmetric(vertical: 16.0),
52         child: ElevatedButton(
53             onPressed: () {
54                 if (_formKey.currentState!.validate()) {
55                     ScaffoldMessenger.of(context).showSnackBar(
56                         SnackBar(content: Text(jsonEncode(formData))));
57                 }
58             },
59             child: Text('Submit'),
60         ),
61     ),
62     ],
63 ),
64 );
65 );
66 }
```

```
export default () => {
  return (
    <View style={styles.container}>
      <Image style={styles.logo} source={require('../assets/hero.jpg')} />
      <Text style={styles.paragraph}>
        Form in React Native, The Right Way!
      </Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    justifyContent: 'center',
    flex: 1,
  },
  paragraph: {
    margin: 24,
    marginTop: 0,
    fontSize: 34,
    fontWeight: 'bold',
    textAlign: 'center',
    color: '#FFF'
  },
  logo: {
    width: '100%',
    height: 200
  }
});

type FormData = {
  name: string;
  email: string;
  password: string;
};

export default () => {
  const { handleSubmit, register, setValue, errors } = useForm<FormData>();

  const onSubmit = (data: FormData) => {
    Alert.alert(`data`, JSON.stringify(data));
  };

  return (
    <KeyboardAwareScrollView>
      <contentContainerStyle={styles.container}>
        <style{`backgroundColor: #181e34`}></style>
        <Form style={styles.formContainer}>
          <Form {...{register, setValue, validation, errors}}>
            <Input name="name" label="Name" />
            <Input name="email" label="Email" />
            <Input name="password" label="Password" secureTextEntry={true} />
            <Button title="Submit" onPress={handleSubmit(onSubmit)} />
          </Form>
        </Form>
      </contentContainerStyle>
    </KeyboardAwareScrollView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    padding: Constants.statusBarHeight,
    backgroundColor: '#181e34',
  },
  formContainer: {
    padding: 8,
    flex: 1,
  },
  button: {
    backgroundColor: 'red',
  },
});
```

```
1 export default {
2   name: { required: true, message: 'Name is required' },
3   email: {
4     required: true,
5     pattern: value => {
6       return /^[a-zA-Z0-9]+([.][a-zA-Z0-9]+)*@[a-zA-Z0-9]+\.[a-zA-Z.]{2,}$/.test(value);
7     },
8     message: 'Invalid Email Format',
9   },
10   password: {
11     required: true,
12     message: 'Password is required',
13   },
14 };
15
16 interface ValidationMap {
17   [key: string]: ValidationOptions;
18 }
19
20 interface ErrorMap {
21   [key: string]: FieldError | undefined;
22 }
23
24 interface Props {
25   children: JSX.Element | JSX.Element[];
26   register: (field: { name: string }, validation?: ValidationOptions) => void;
27   errors: ErrorMap;
28   validation: ValidationMap;
29   setValue: (name: string, value: string, validate?: boolean) => void;
30 }
31
32 export default ({  
33   register,  
34   errors,  
35   setValue,  
36   validation,  
37   children,  
38 }: Props) => {  
39   const Inputs = React.useRef<Array<TextInput>>([]);  
40  
41   React.useEffect(() => {  
42     (Array.isArray(children) ? [...children] : [children]).forEach((child) => {  
43       if (child.props.name)  
44         register({ name: child.props.name }, validation[child.props.name]);  
45     });
46   }, [register]);  
47  
48   return (  
49     <>  
50       (Array.isArray(children) ? [...children] : [children]).map(  
51         (child, i) => {  
52           return child.props.name  
53             ? React.createElement(child.type, {  
54               ...child.props,  
55               ref: (e: TextInput) => {  
56                 Inputs.current[i] = e;  
57               },  
58               onChangeText: (v: string) =>  
59                 setValue(child.props.name, v, true),  
60               onSubmitEditing: () => {  
61                 Inputs.current[i + 1]  
62                   ? Inputs.current[i + 1].focus()  
63                   : Inputs.current[i].blur();  
64               },  
65               onBlur: () => triggerValidation(child.props.name),  
66               blurOnSubmit: false,  
67               name: child.props.name,  
68               error: errors[child.props.name],  
69             },  
70           ),  
71         ),  
72       )  
73     )
74   );
75 };
76 </>
77 );
78
79 );
```

```
19 export default React.forwardRef<any, Props>
20   (props, ref) => React.createElement <>
21     const { label, labelStyle, error, ...inputProps } = props;
22
23     return (
24       <View style={styles.container}>
25         <Label>{label}</Label>
26         <Text style={[styles.label, labelStyle]}>{label}</Text>
27         <TextInput
28           autoCapitalize="none"
29           ref={ref}
30           style={[styles.input, { borderColor: error ? '#fc6d47' : '#c0cb3d' }]}
31           {...inputProps}
32         />
33         <Text style={styles.textError}>{error && error.message}</Text>
34       </View>
35     );
36   );
37
38 const styles = StyleSheet.create({
39   container: {
40     marginVertical: 8,
41   },
42   input: {
43     borderStyle: 'solid',
44     borderWidth: 1,
45     borderRadius: 5,
46     paddingVertical: 5,
47     paddingRight: 5,
48     height: 40,
49     color: '#c0cb3d',
50   },
51   label: {
52     paddingVertical: 5,
53     fontSize: 16,
54     fontWeight: 'bold',
55     color: '#c0cb3d',
56   },
57   textError: {
58     color: '#fc6d47',
59     fontSize: 14,
60   },
61 });
62 );
```

```
19 export default React.forwardRef<any, Props>(
20   (props, ref): React.ReactElement => {
21     const { label, labelStyle, error, ...inputProps } = props;
22
23     return (
24       <View style={styles.container}>
25         {label && <Text style={[styles.label, labelStyle]}>{label}</Text>}
26         <TextInput
27           autoCapitalize="none"
28           ref={ref}
29           style={[styles.input, { borderColor: error ? '#fc6d47' : '#c0cbd3' }]}
30           {...inputProps}
31         />
32         <Text style={styles.textError}>{error && error.message}</Text>
33       </View>
34     );
35   }
36 );
37
38 const styles = StyleSheet.create({
39   container: {
40     marginVertical: 8,
41   },
42   input: {
43     borderStyle: 'solid',
44     borderWidth: 1,
45     borderRadius: 5,
46     paddingVertical: 5,
47     paddingLeft: 5,
48     fontSize: 16,
49     height: 40,
50     color: '#c0cbd3',
51   },
52   label: {
53     paddingVertical: 5,
54     fontSize: 16,
55     fontWeight: 'bold',
56     color: '#c0cbd3',
57   },
58   textError: {
59     color: '#fc6d47',
60     fontSize: 14,
61   },
62 });
63 
```

```
14
15 const styles = StyleSheet.create({
16   container: {
17     justifyContent: 'center',
18     flex:1,
19   },
20   paragraph: {
21     margin: 24,
22     marginTop: 0,
23     fontSize: 34,
24     fontWeight: 'bold',
25     textAlign: 'center',
26     color: '#FFF'
27   },
28   logo: {
29     width: '100%',
30     height: 200
31   }
32 });
```



| | | | |
|--------------|--------------|-----------------|------------|
| Flutter | Programmcode | main.dart | 60 |
| | | validation.dart | 15 |
| | | Summe | 75 |
| React Native | Programmcode | App.tsx | 15 |
| | | Hero.tsx | 9 |
| | | Input.tsx | 17 |
| | | Form.tsx | 52 |
| | | validation.tsx | 12 |
| | | Summe | 112 |
| | Stylesheets | App.tsx | 14 |
| | | Hero.tsx | 17 |
| | | Input.tsx | 24 |
| | | Summe | 55 |
| Summe | | 167 | |

Welche Technologie ?



Problem #1

Um im Auswahlfeld der Zieldaten die Option `md` auswählen zu können muss:

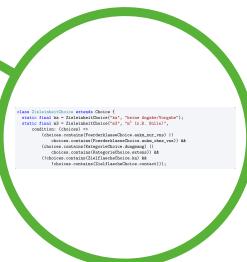
- für die `ProductKlasse` entweder
 - Argument und Kontrahenten; nur Wehrgegenabschuss oder
 - Argument und Kontrahenten, die auch mit Tewerkasperien, aber ohne Wehrgegenabschuss
- und für die `Kategorie` entweder
 - Diversifikation oder
 - Erweiterung
- und für Ziellösste mit weder
 - kein Argon, noch
 - alle um Unterstützung geeignet sein.

```
class Diversifikationsklasse extends Choice {  
    static final String NAME = "Diversifikationsklasse";  
    static final String ARGUMENT = "wehrgegenabschuss";  
    static final String CONTRAHENTEN = "keine wehrgegenabschuss";  
    static final String KATEGORIE = "diversifikation";  
    static final String ZIELLOESSTE = "keine wehrgegenabschuss";  
    static final String WEHRGEGENABSCHUSS = "wehrgegenabschuss";  
    static final String ERWEITERUNG = "erweiterung";  
    static final String DIVERSIFIKATION = "diversifikation";  
    static final String KONTRAHENTEN = "keine wehrgegenabschuss";  
    static final String ARGON = "kein argon";  
    static final String UNTERSTUETZUNG = "unterstuetzung";  
    static final String KEINE_UNTERSTUETZUNG = "keine unterstuetzung";  
    static final String WEHRGEGENABSCHUSS_KONTRAHENTEN = "wehrgegenabschuss_kontrahenten";  
    static final String WEHRGEGENABSCHUSS_ARGON = "wehrgegenabschuss_argon";  
    static final String ERWEITERUNG_KONTRAHENTEN = "erweiterung_kontrahenten";  
    static final String DIVERSIFIKATION_KONTRAHENTEN = "diversifikation_kontrahenten";  
    static final String DIVERSIFIKATION_ARGON = "diversifikation_argon";  
    static final String ERWEITERUNG_ARGON = "erweiterung_argon";  
    static final String DIVERSIFIKATION_UNTERSTUETZUNG = "diversifikation_unterstuetzung";  
    static final String ERWEITERUNG_UNTERSTUETZUNG = "erweiterung_unterstuetzung";  
    static final String DIVERSIFIKATION_KONTRAHENTEN_UNTERSTUETZUNG = "diversifikation_kontrahenten_unterstuetzung";  
    static final String ERWEITERUNG_KONTRAHENTEN_UNTERSTUETZUNG = "erweiterung_kontrahenten_unterstuetzung";  
}
```

Um im Auswahlfeld der *Zieleinheit* die Option *m3* auswählen zu können muss:

- für die *Förderklasse* entweder
 - *Agrarumwelt-(und Klima)Maßnahme: nur Vertragsnaturschutz oder*
 - *Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz*
- **und** für die *Kategorie* entweder
 - *Düngemanagement oder*
 - *Extensivierung*
- **und** für *Zielfläche* darf **weder**
 - *keine Angabe, noch*
 - *bitte um Unterstützung gewählt sein.*

Problem #1



```
class ZieleinheitChoice extends Choice {  
    static final ka = ZieleinheitChoice("ka", "keine Angabe/Vorgabe");  
    static final m3 = ZieleinheitChoice("m3", "m3 (z.B. Gülle)",  
        condition: (choices) =>  
            (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||  
                choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&  
            (choices.contains(KategorieChoice.dungmang) ||  
                choices.contains(KategorieChoice.extens)) &&  
            (!choices.contains(ZielflaecheChoice.ka) &&  
                !choices.contains(ZielflaecheChoice.contact)));
```

Problem # 1

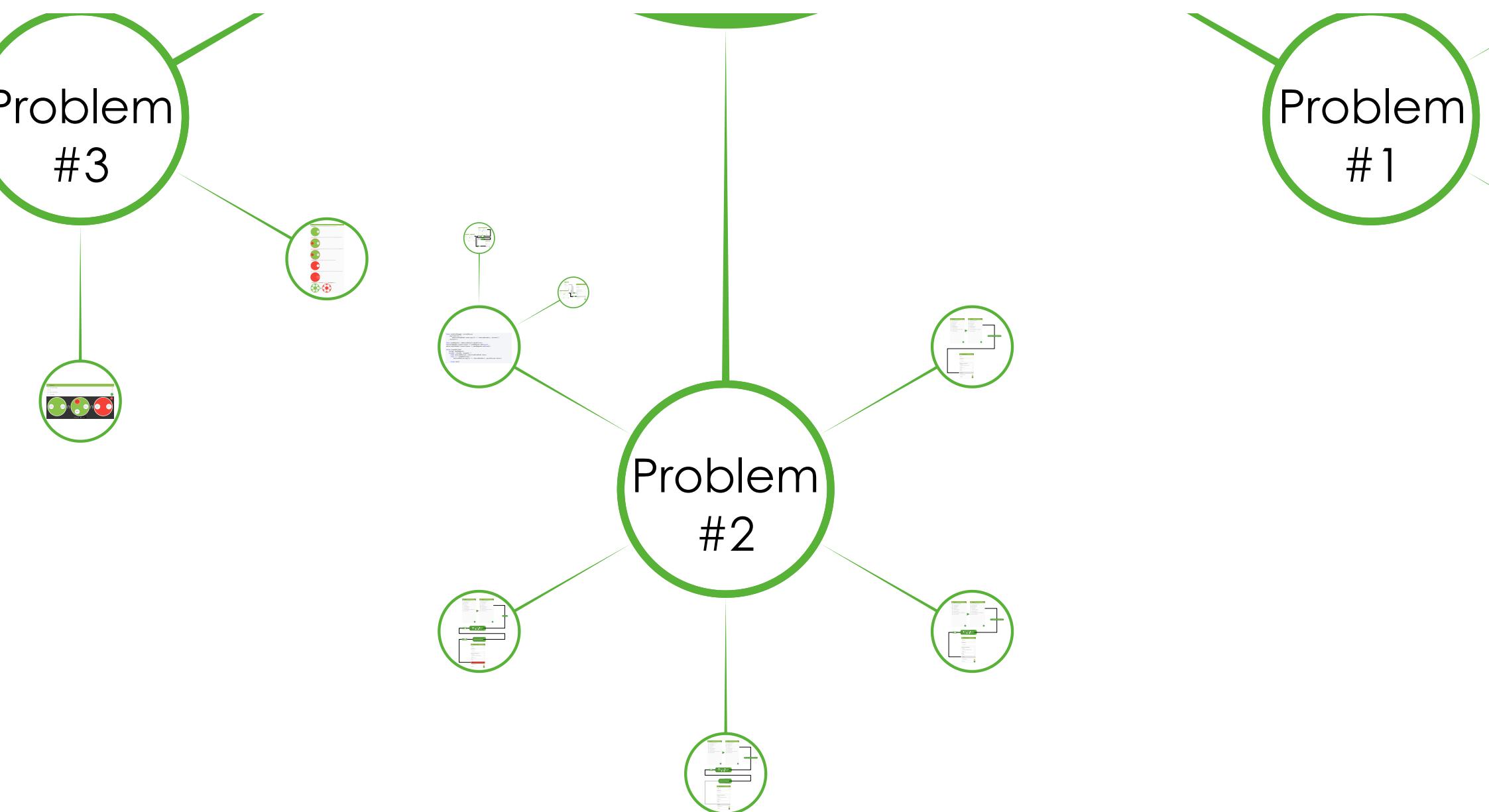
Die in Ausweitung der Zielsetzung mit Zusatzfunktion zu können muss:

- für die Filterfunktionen ersterer
- Aggregation und Konditionierung, nur Wirtschaftlichkeit oder
- Aggregation und Konditionierung, die auch mit Sonderkosten oder ohne Wirtschaftlichkeit
- und für die Kategorien ersterer
- Aggregation und
- Konditionierung
- und für Zweitfunktionen ersterer
- keine Angabe noch
- dann eine Unterscheidung gewünscht wird.

Die Funktionen müssen darin, entweder:

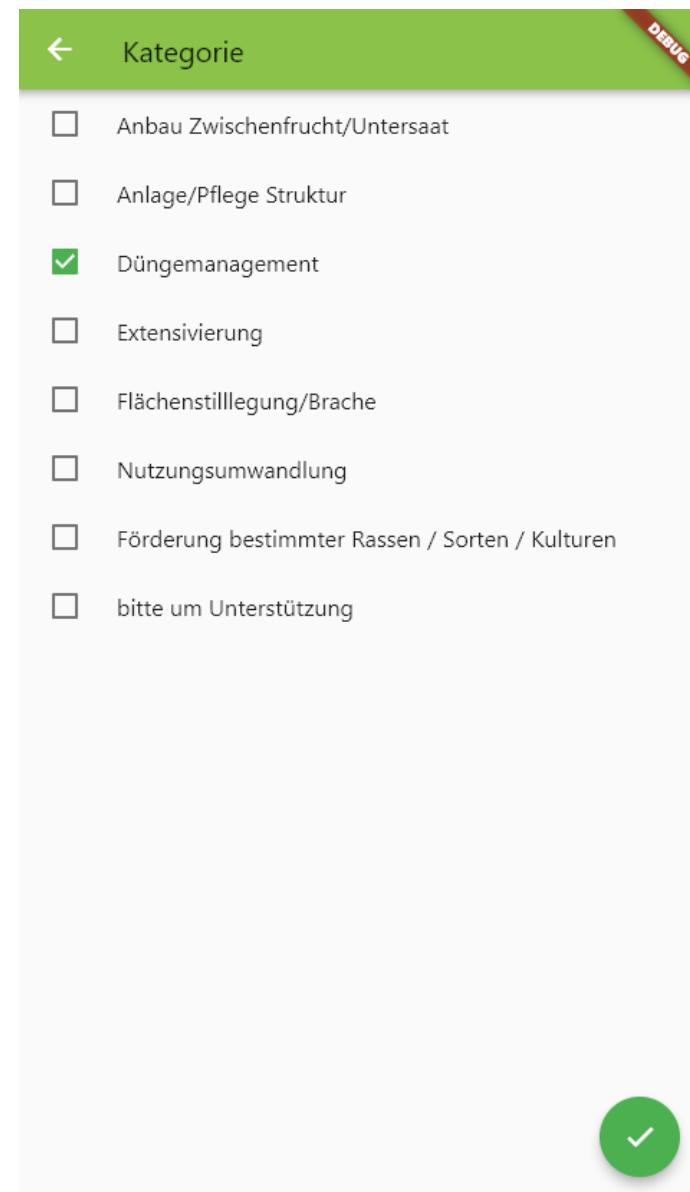
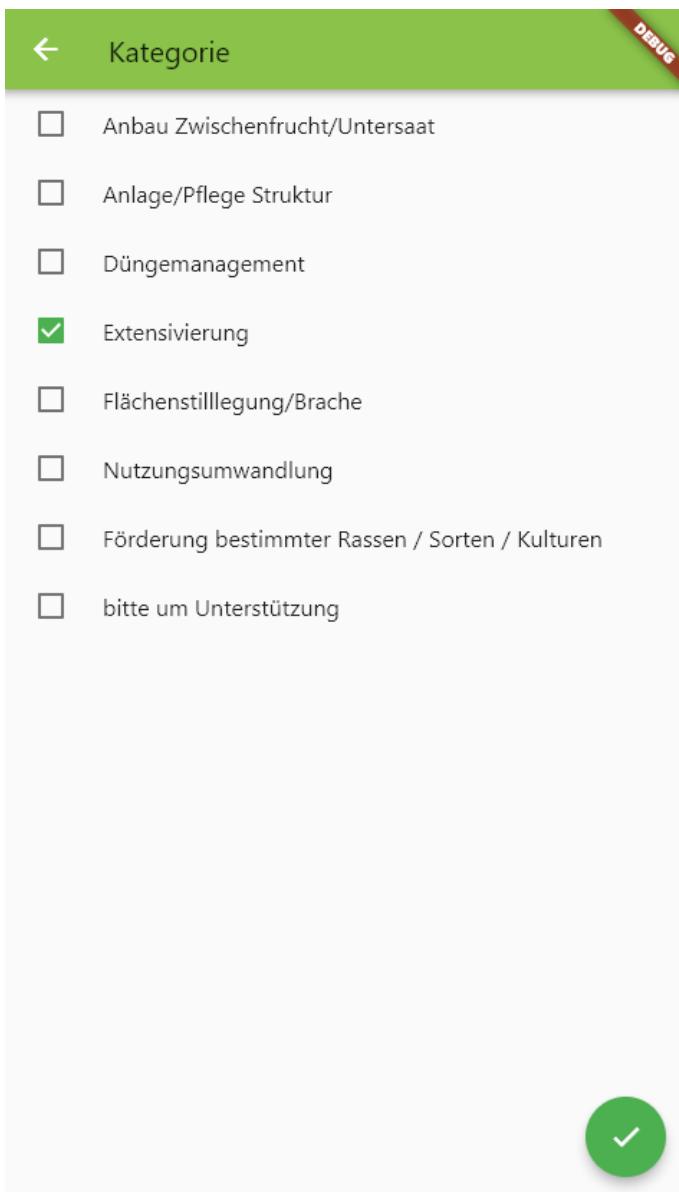
- ersten kann es + Einstellungsfunktionen, "Welche Kapazität", "Welche Anzahl", "Welche Kosten"
- zweitens: "Geben Sie mir die entsprechenden Werte aus der Zeit 11"
- drittens: "Geben Sie mir die entsprechenden Werte aus der Zeit 12"
- viertens: "Geben Sie mir die entsprechenden Werte aus der Zeit 13"
- fünftens: "Geben Sie mir die entsprechenden Werte aus der Zeit 14"
- sechstens: "Geben Sie mir die entsprechenden Werte aus der Zeit 15"



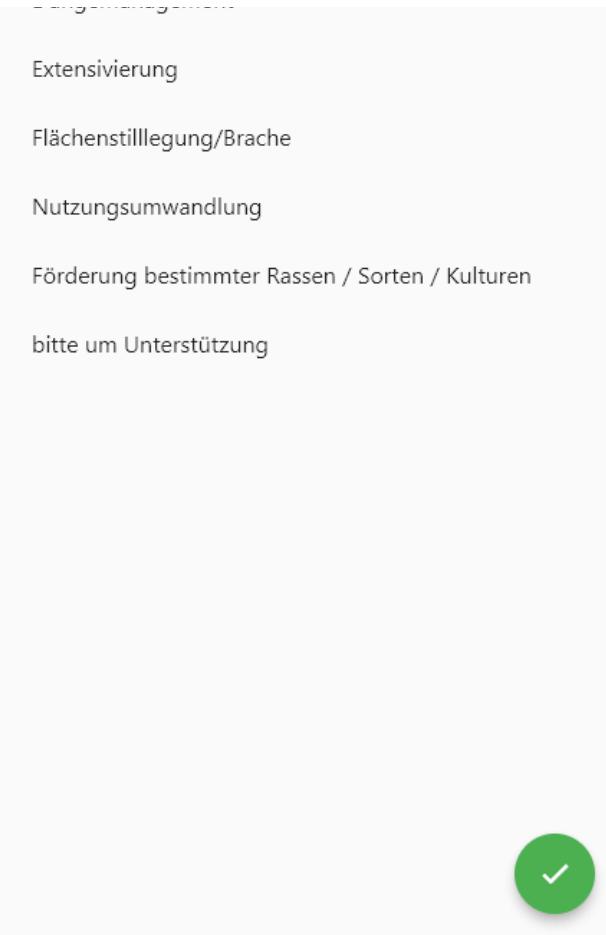


The image shows a user interface for managing agricultural measures. It consists of three main parts:

- Screenshot 1 (Left):** A list of categories with checkboxes. The "Extensivierung" checkbox is checked. Other options include Anbau Zwischenfrucht/Untersaat, Anlage/Pflege Struktur, Düngemanagement, Flächenbearbeitung/Brache, Nutzungsumwandlung, Förderung bestimmter Rassen / Sorten / Kulturen, and bitte um Unterstützung.
- Screenshot 2 (Right):** A similar list of categories. The "Düngemanagement" checkbox is checked. Other options are the same as in Screenshot 1.
- Bottom Screenshot:** A detailed view of a measure. It includes fields for Status (in Bearbeitung), Identifikatoren (Maßnahmetitel), Maßnahmencharakteristika (Förderklasse: Agrarumwelt- und KlimaMaßnahmen, tv. auch mit Themenaspekten alter ORNL Vertragsnaturrauschutz; Kategorie: Düngemanagement), Zielsetzung (Zielfläche, Zielinhalt: z.B. Gras), Hauptzielbesetzung Land (marked with an orange square), and Nebenziele (marked with a green checkmark).



DÜ



Düngemanagement

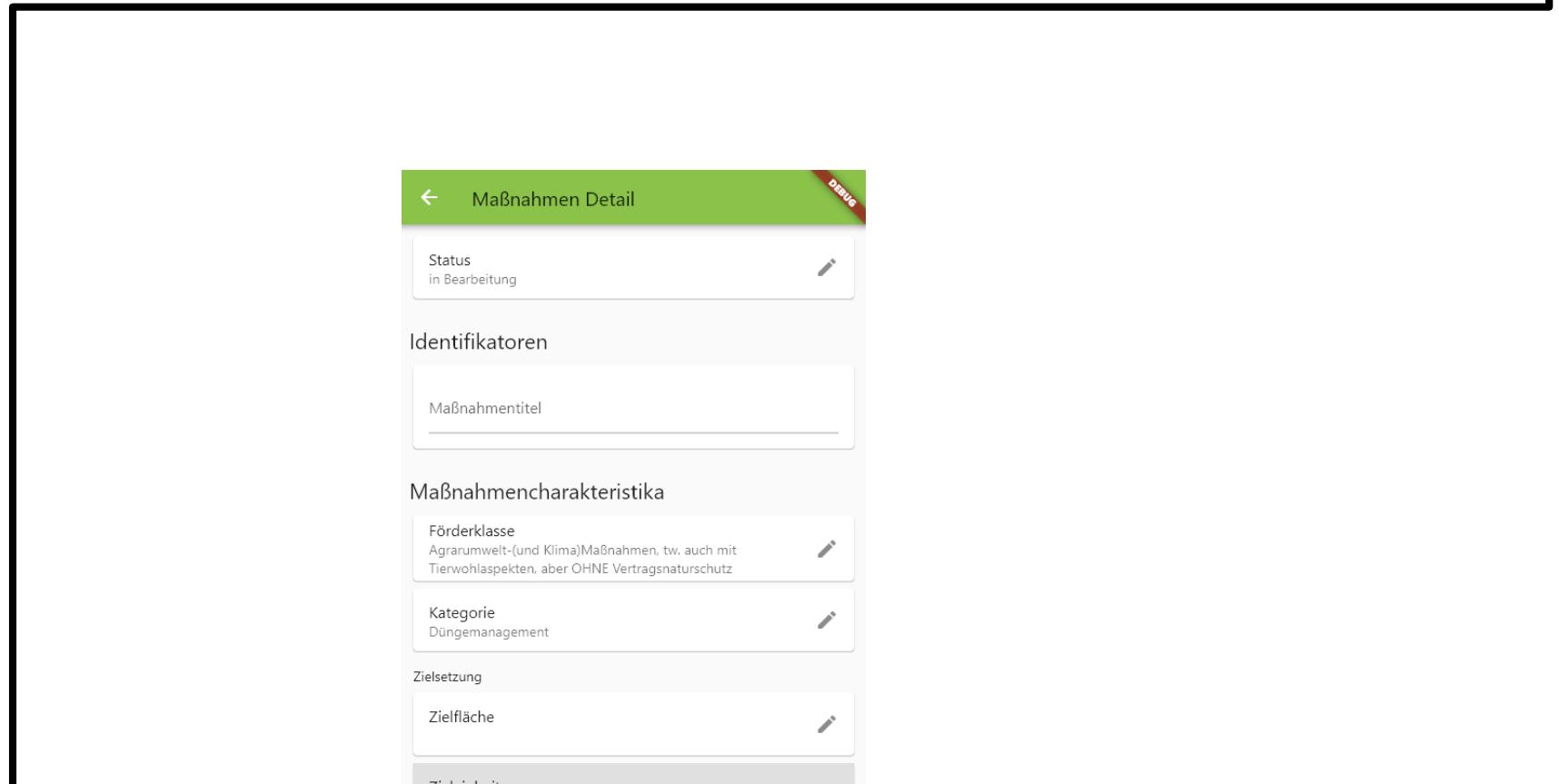
Extensivierung

Flächenstilllegung/Brache

Nutzungsumwandlung

Förderung bestimmter Rassen / Sorten / Kulturen

bitte um Unterstützung



Kategorie

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement **(checked)**
- Extensivierung
- Flächenstiftung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

Kategorie

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement **(checked)**
- Extensivierung
- Flächenstiftung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

Düngemanagement

Maßnahmen Detail

Status
in Bearbeitung

Identifikatoren

Maßnahmendst

Maßnahmencharakterika

Förderklasse
Agrarwende- und KlimaMaßnahmen, zw. auch mit
Wirtschaftsgütern aller CHNE Beitragsträger

Kategorie
Düngemanagement

Zielsetzung

Zielfläche

Zielheit
soz. (z.B. Güte)

Hauptzielsetzung Land

Nebenziele



← Kategorie DEBUG

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



← Kategorie DEBUG

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



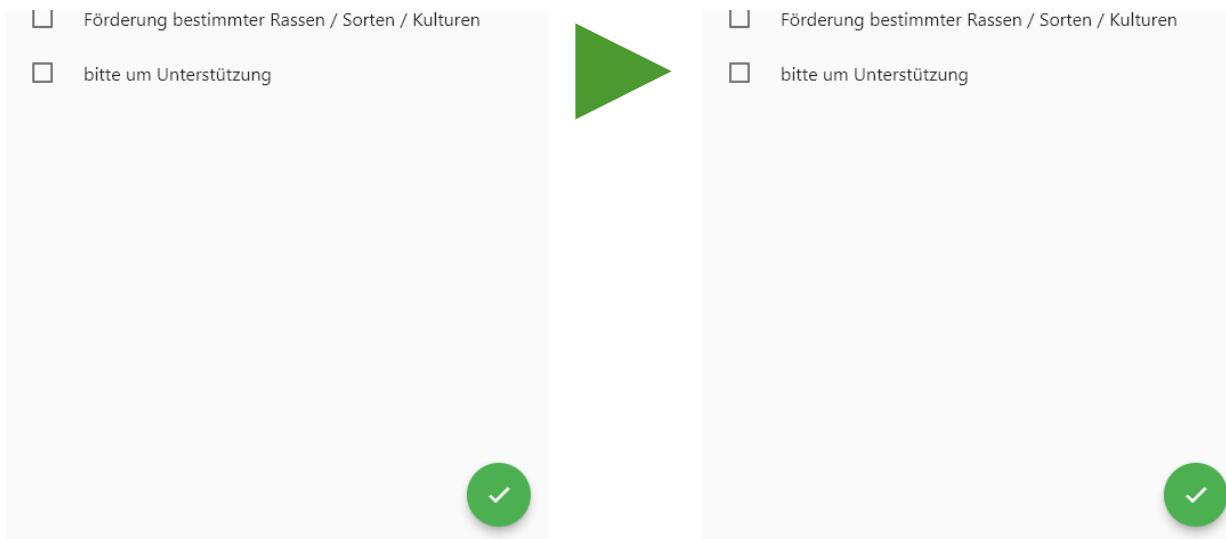
Dü

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



Düngemanagement



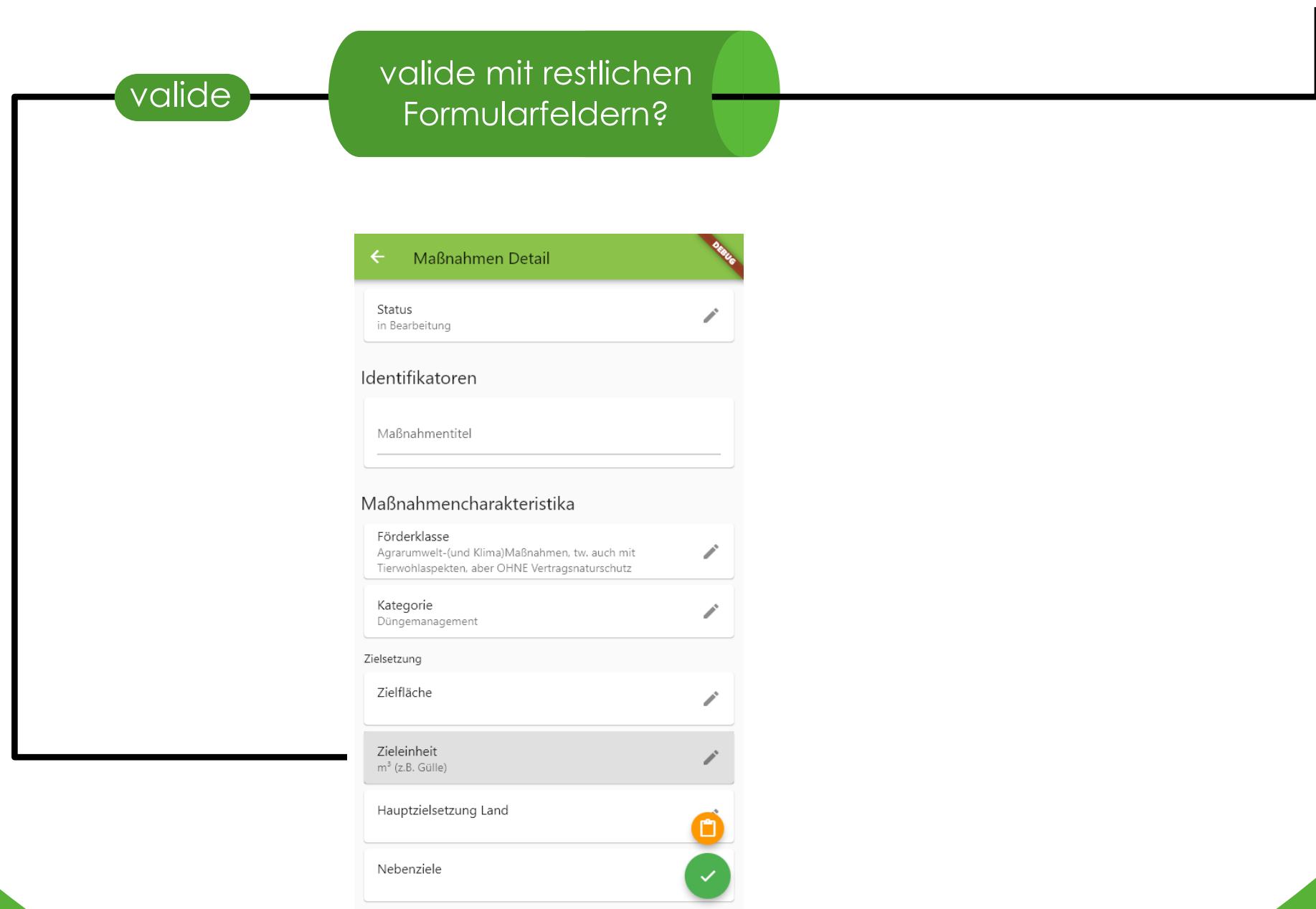


Düngemanagement

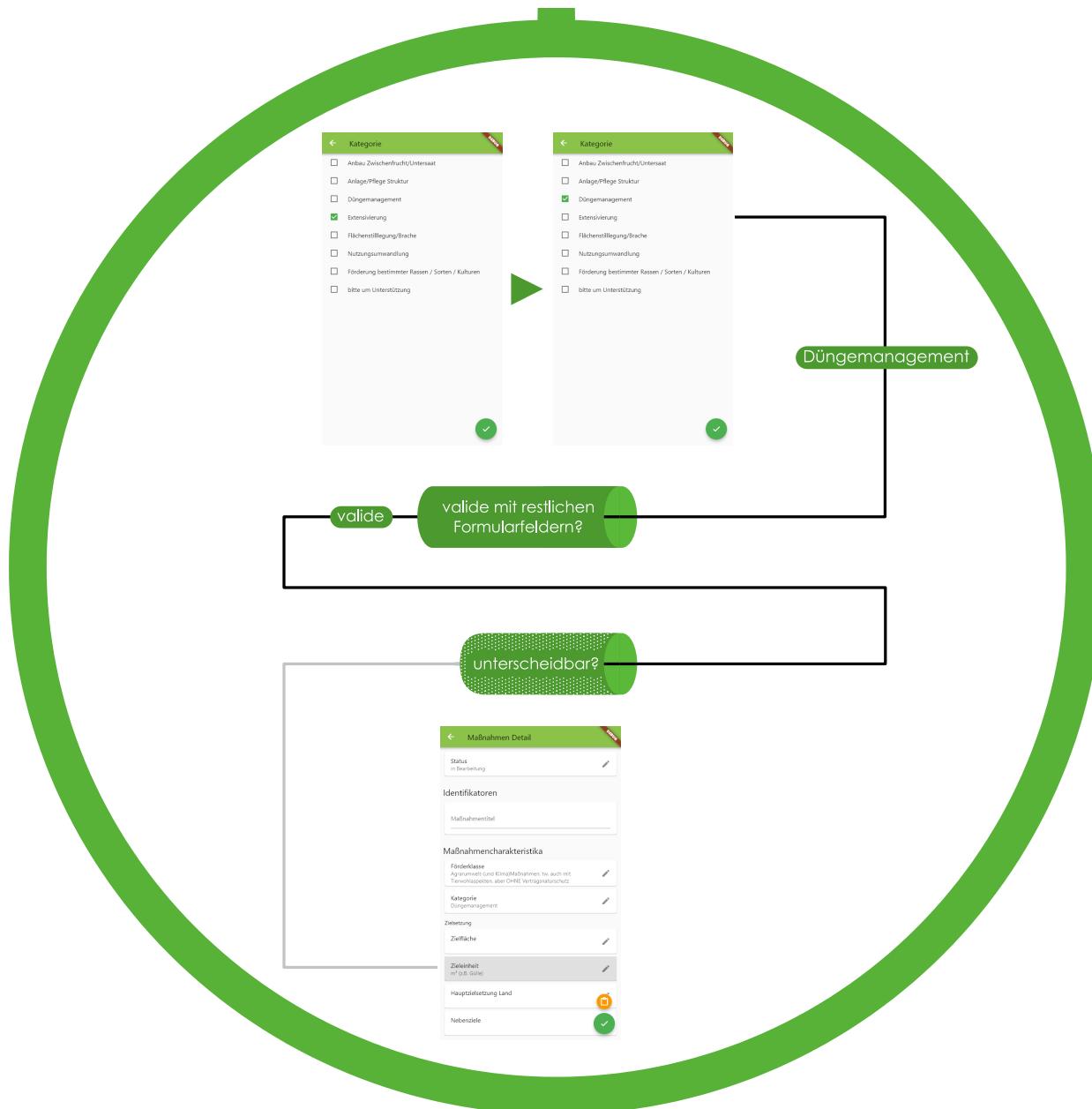
valide

valide mit restlichen
Formularfeldern?

The image shows a screenshot of a mobile application screen titled "Maßnahmen Detail". At the top, there is a navigation bar with a back arrow and the title "Maßnahmen Detail". To the right of the title, there is a small red tag with the word "DEBUG". Below the navigation bar, the screen displays the status "Status in Bearbeitung" and a "Identifikatoren" section. Within the "Identifikatoren" section, there is a field labeled "Maßnahmentitel".







Kategorie

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

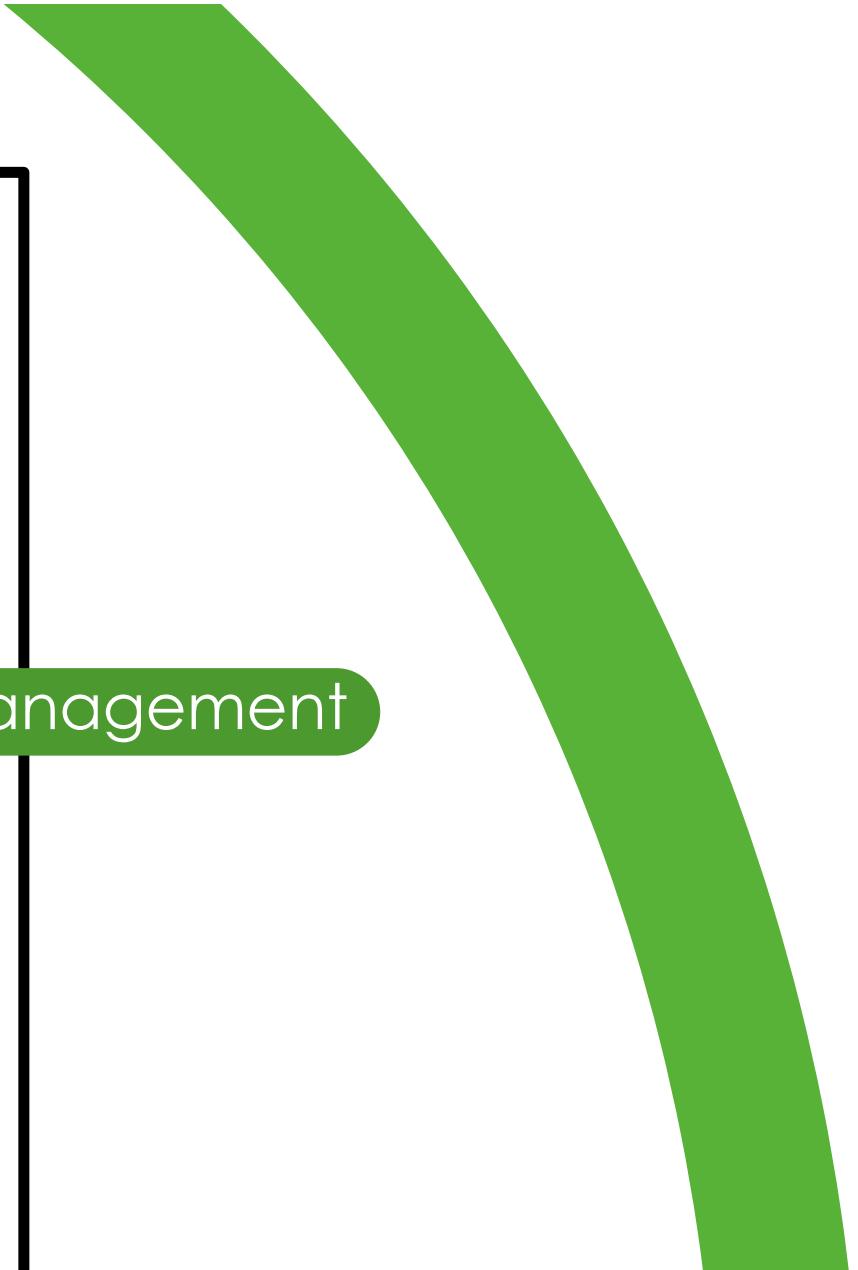
Kategorie

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



DÜ

- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



Düngemanagement



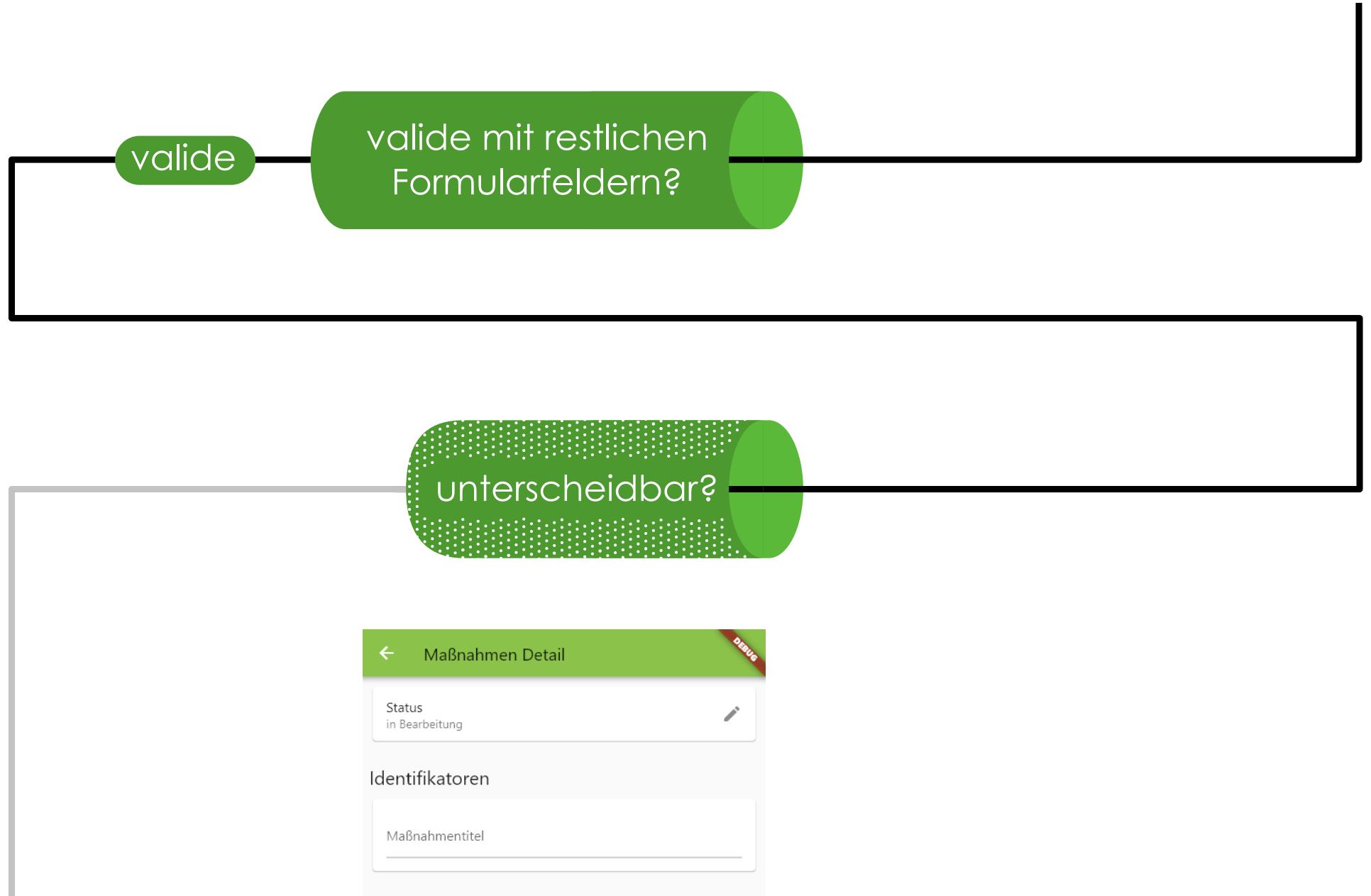
Düngemanagement



valide

valide mit restlichen
Formularfeldern?

unterscheidbar?





← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Düngemanagement

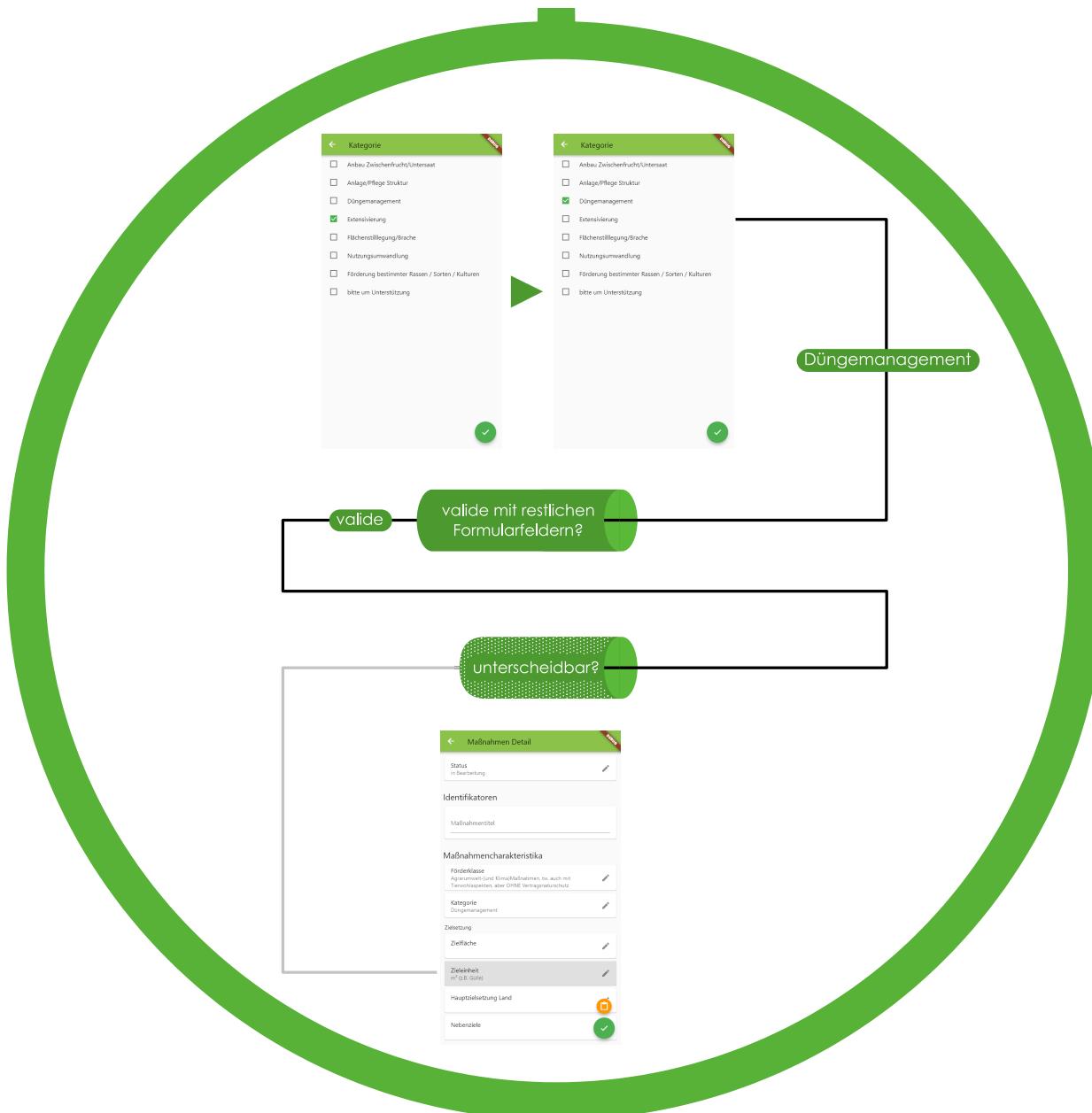
Zielsetzung

Zielfläche

Zieleinheit
 m^3 (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele





← Kategorie DEBUG

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

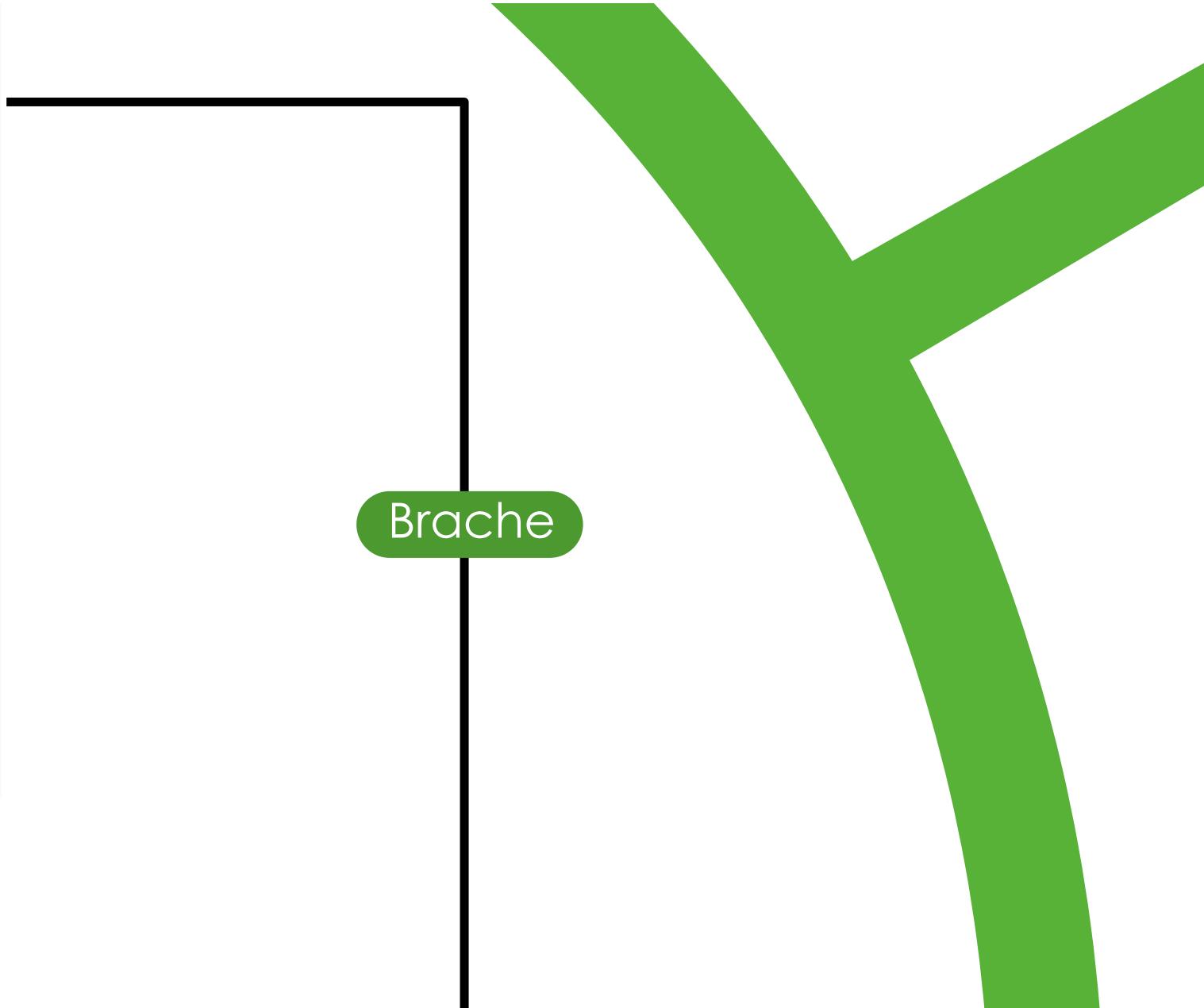


← Kategorie DEBUG

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



A large, stylized green checkmark graphic is positioned on the right side of the slide. It consists of a thick green V-shape pointing upwards and to the left, intersected by a vertical black line that extends from the bottom of the list area up to the word "Brache".

Brache



A horizontal green bar with a rounded end, located at the bottom left of the slide.

tlichen

- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

Brache



invalid

valide mit restlichen
Formularfeldern?



invalid

unterscheidbar?

← Maßnahmen Detail DEU

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

Kategorie
Flächenstilllegung/Brache

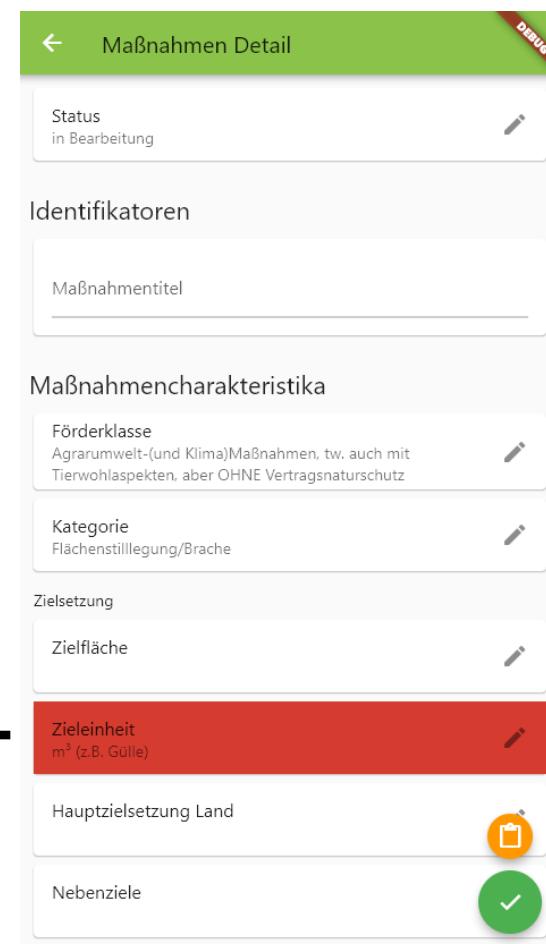
Zielsetzung

Zielfläche

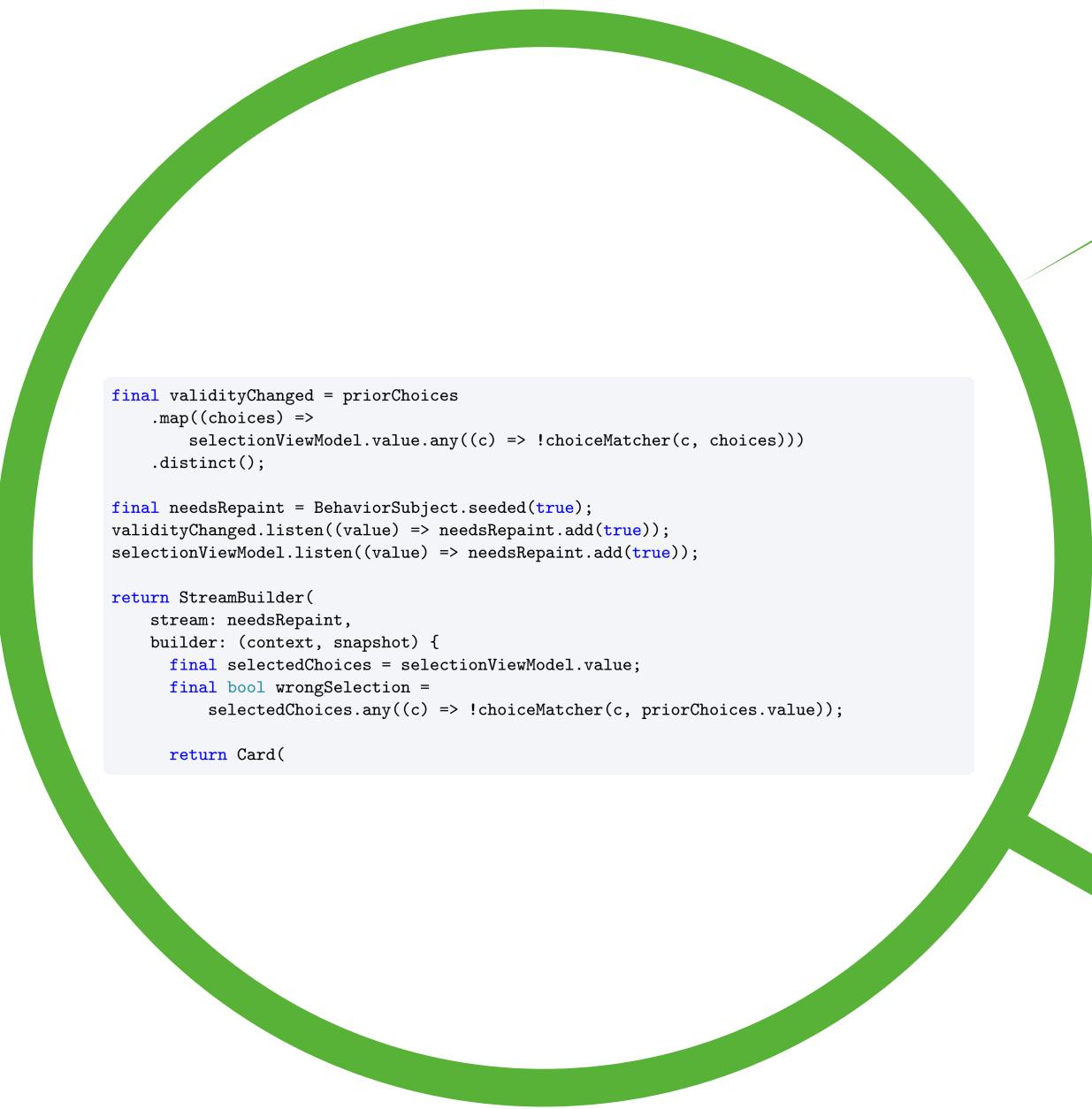
Zieleinheit
m³ (z.B. Gülle)

Hauptzielsetzung Land

Nebenziele







```
final validityChanged = priorChoices
    .map((choices) =>
        selectionViewModel.value.any((c) => !choiceMatcher(c, choices)))
    .distinct();

final needsRepaint = BehaviorSubject.seeded(true);
validityChanged.listen((value) => needsRepaint.add(true));
selectionViewModel.listen((value) => needsRepaint.add(true));

return StreamBuilder(
    stream: needsRepaint,
    builder: (context, snapshot) {
        final selectedChoices = selectionViewModel.value;
        final bool wrongSelection =
            selectedChoices.any((c) => !choiceMatcher(c, priorChoices.value));

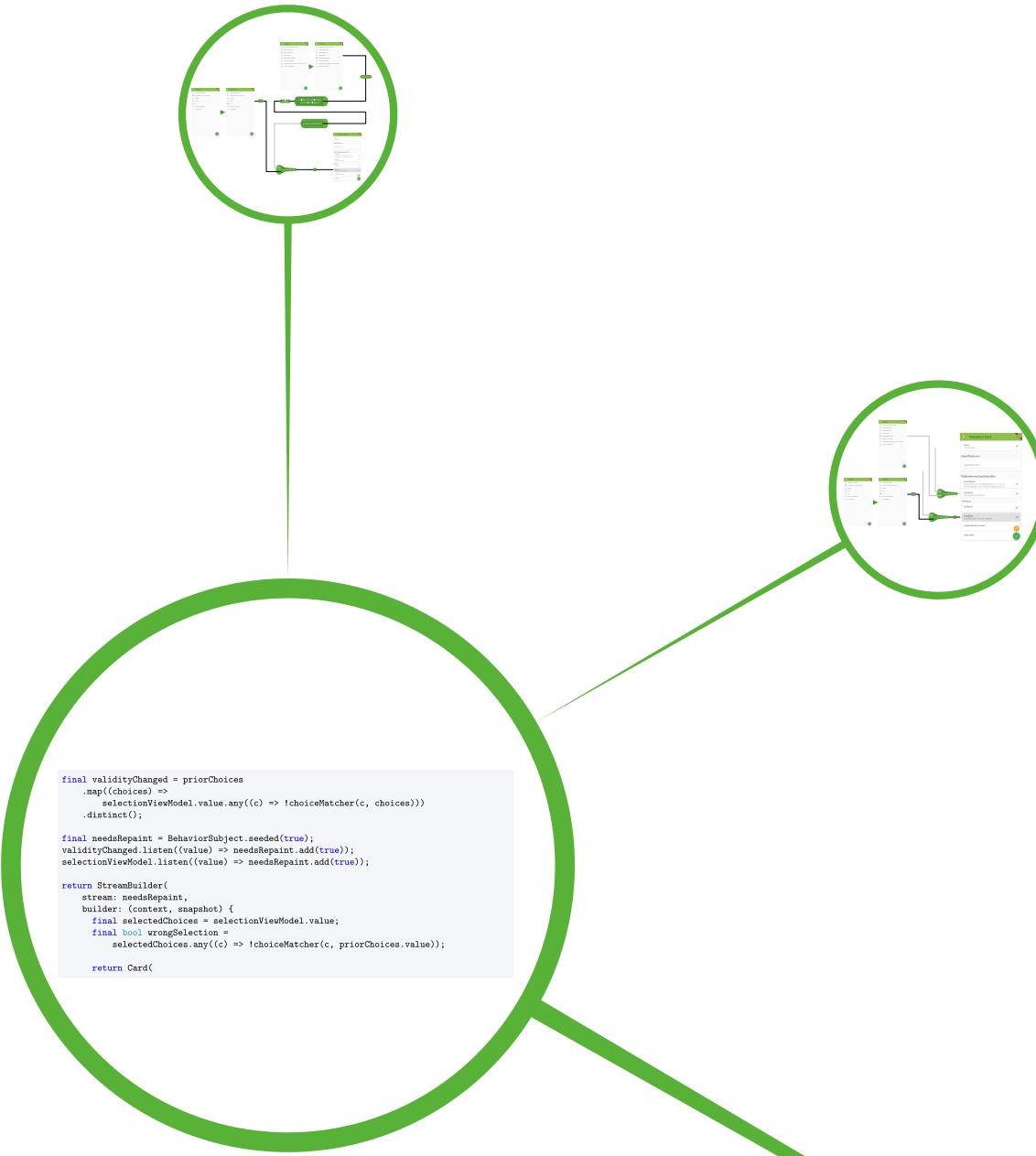
        return Card(
```

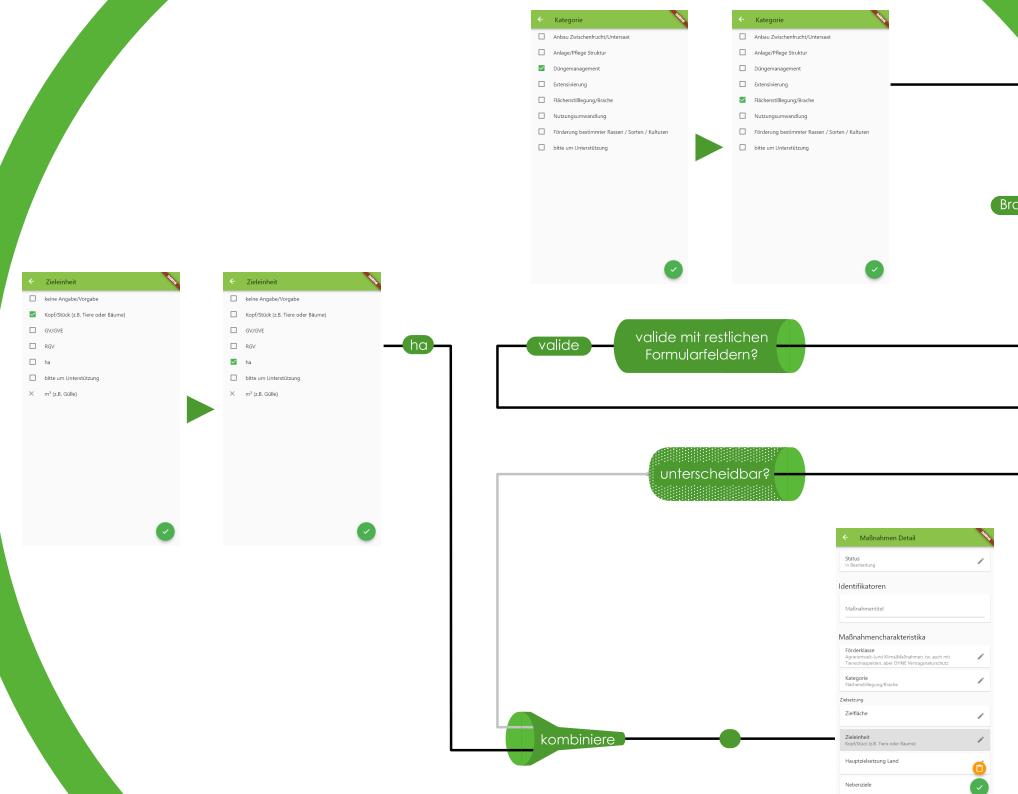
```
final validityChanged = priorChoices
    .map((choices) =>
        selectionViewModel.value.any((c) => !choiceMatcher(c, choices)))
    .distinct();

final needsRepaint = BehaviorSubject.seeded(true);
validityChanged.listen((value) => needsRepaint.add(true));
selectionViewModel.listen((value) => needsRepaint.add(true));

return StreamBuilder(
    stream: needsRepaint,
    builder: (context, snapshot) {
        final selectedChoices = selectionViewModel.value;
        final bool wrongSelection =
            selectedChoices.any((c) => !choiceMatcher(c, priorChoices.value));

        return Card(
```





← Zieleinheit

DEBÜS

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)

 Kategorie 

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



The image displays two screenshots of a mobile application interface, likely a checklist or survey tool. Both screens have a header bar at the top with a back arrow, the word "Kategorie", and a red "DEBUG" badge.

Left Screen:

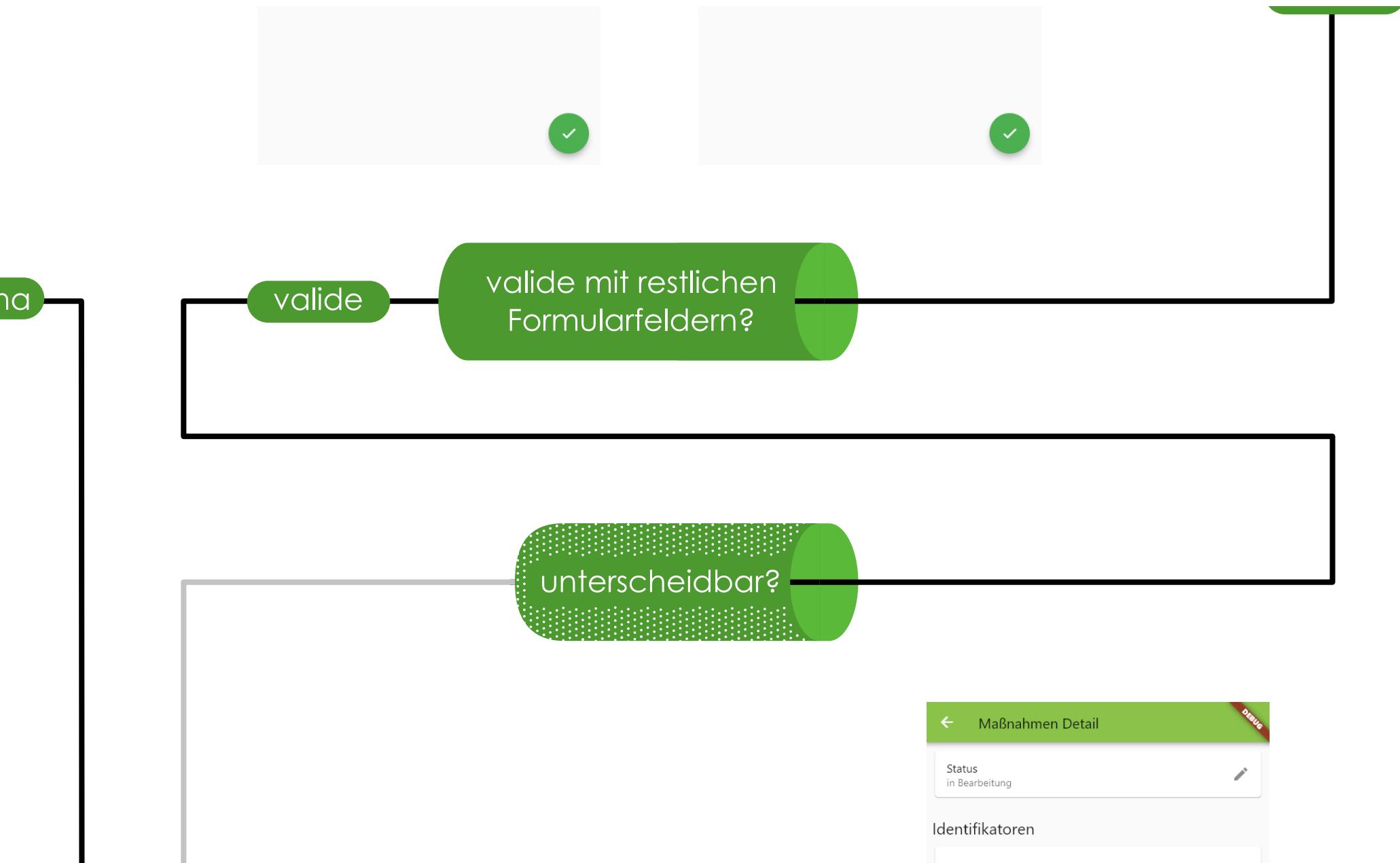
- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

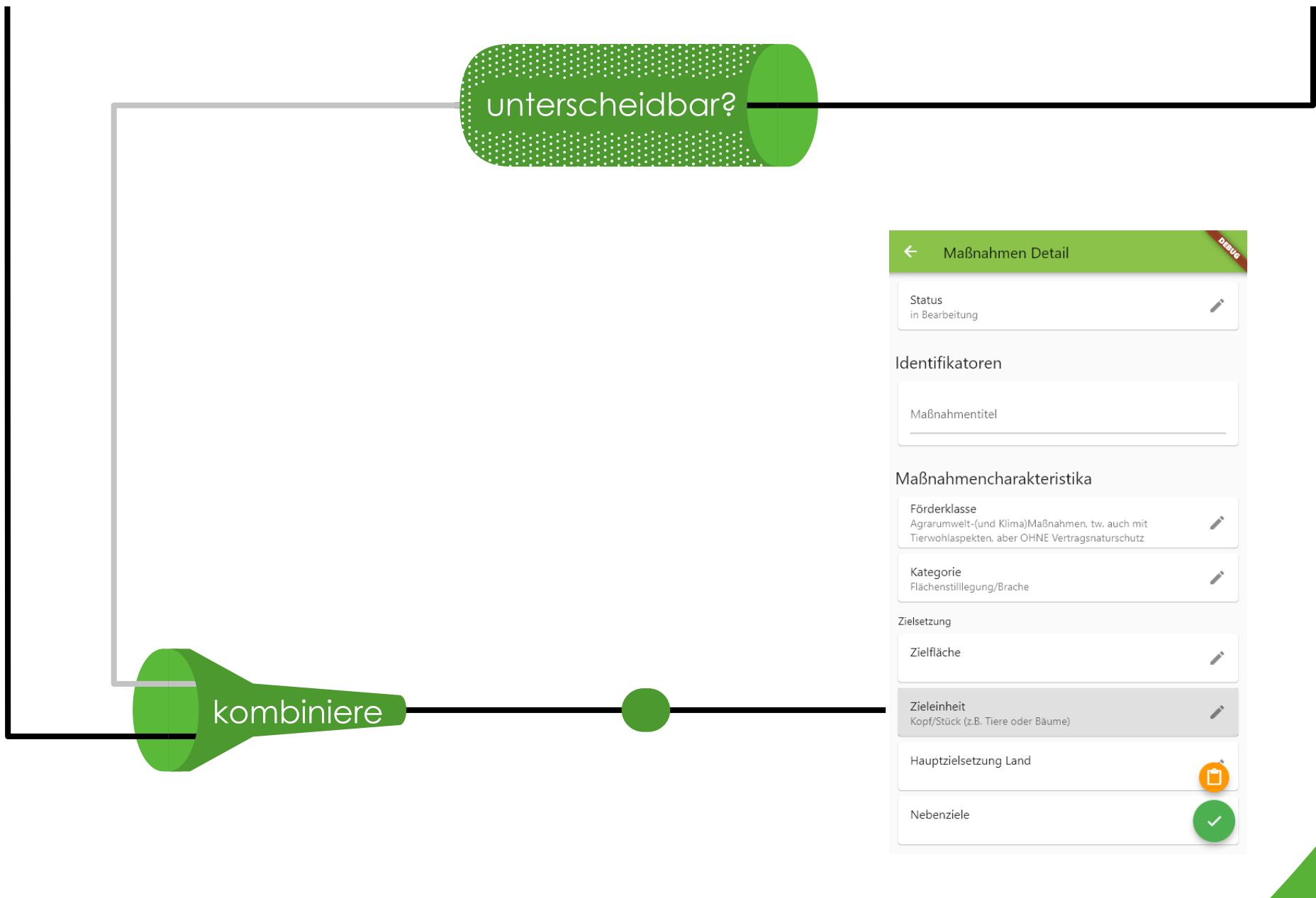
Right Screen:

- Anbau Zwischenfrucht/Untersaat
- Anlage/Pflege Struktur
- Düngemanagement
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung

A large green triangle pointing right is positioned between the two screens, indicating a transition or comparison between the two states. At the bottom center of each screen is a green circle containing a white checkmark.







← Zieleinheit DEBUG

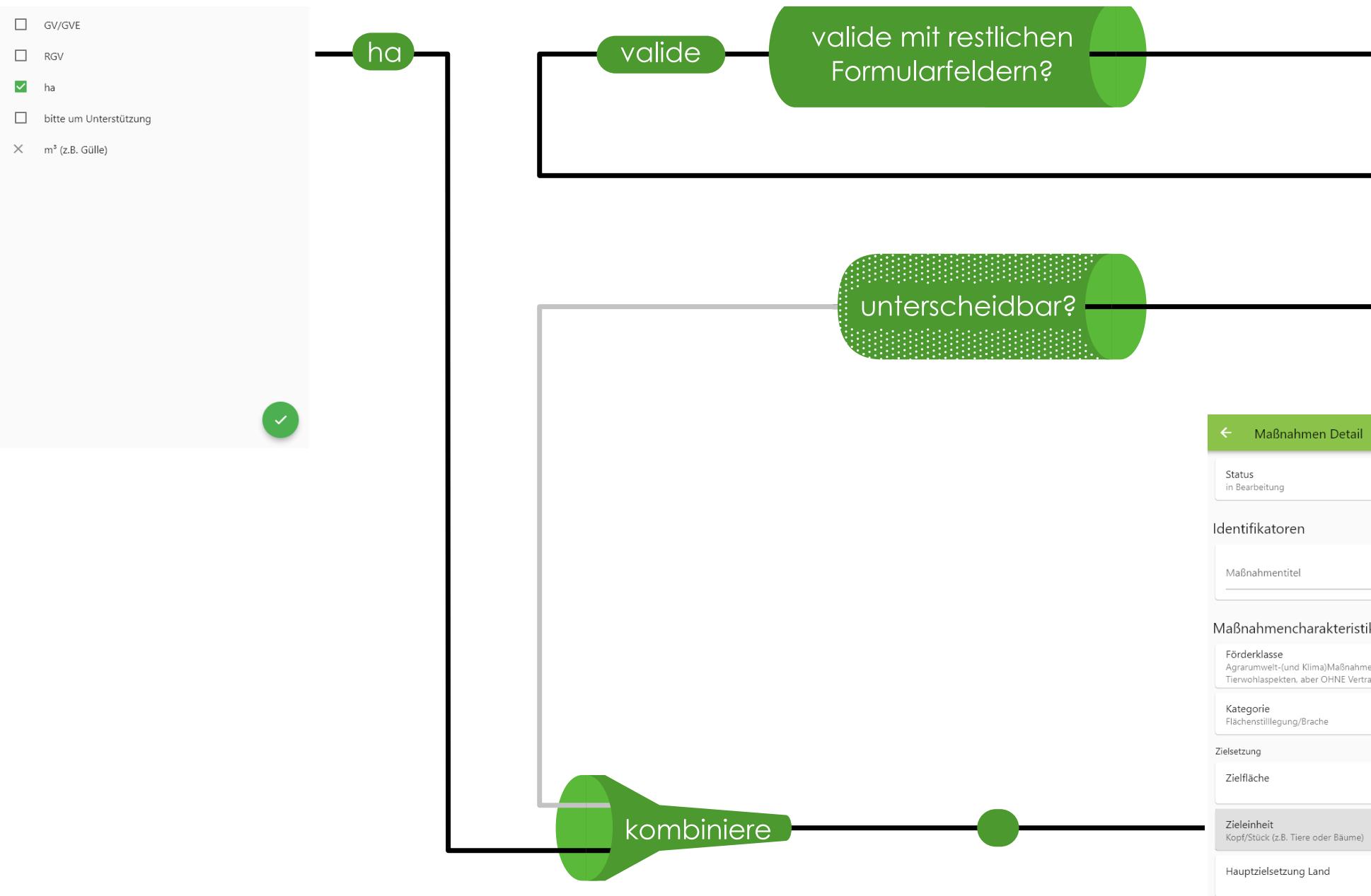
- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)



← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)







← Maßnahmen Detail DEBUG

Status
in Bearbeitung pen

Identifikatoren

Maßnahmentitel pen

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz pen

Kategorie
Flächenstilllegung/Brache pen

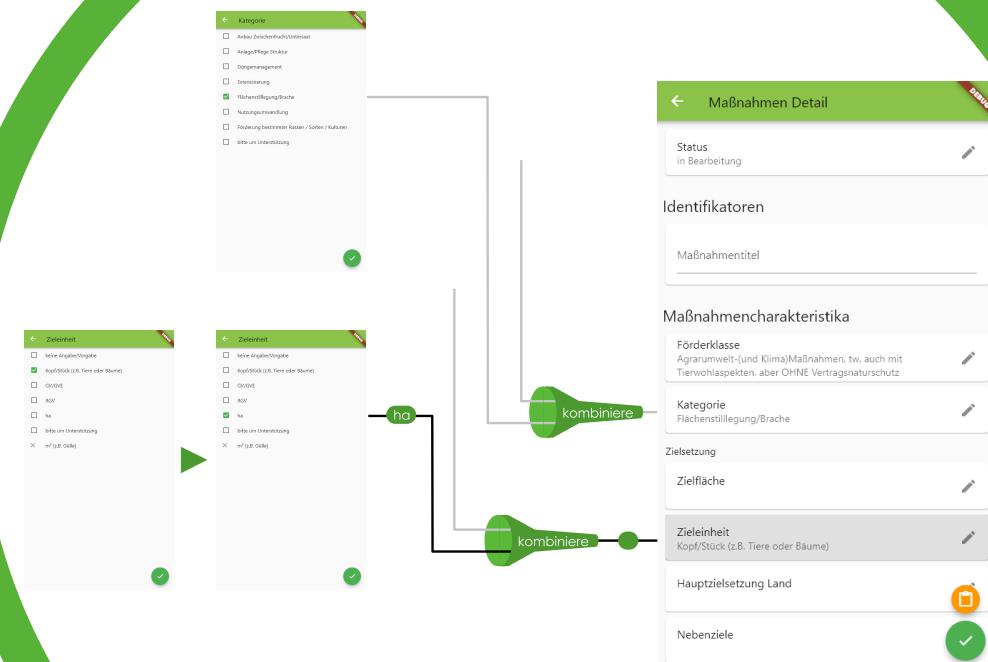
Zielsetzung

Zielfläche pen

Zieleinheit
Kopf/Stück (z.B. Tiere oder Bäume) pen

Hauptzielsetzung Land camera

Nebenziele checkmark



← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)

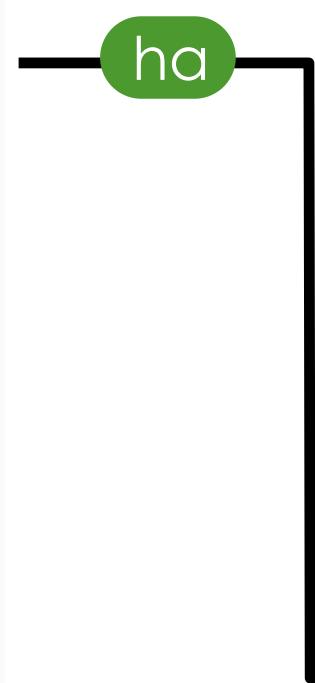




← Zieleinheit DEBUG

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)





The screenshot shows a software application with a flowchart on the left and a configuration panel on the right.

Flowchart:

- A green oval node labeled "ha" is connected to a black vertical line.
- The black vertical line branches into two paths:
 - One path goes up to another green oval node labeled "kombiniere".
 - The other path goes down to a second green oval node labeled "kombiniere".
- Both "kombiniere" nodes are connected to a single green circular end node.

Configuration Panel:

Maßnahmencharakteristika

Förderklasse: Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz (edit icon)

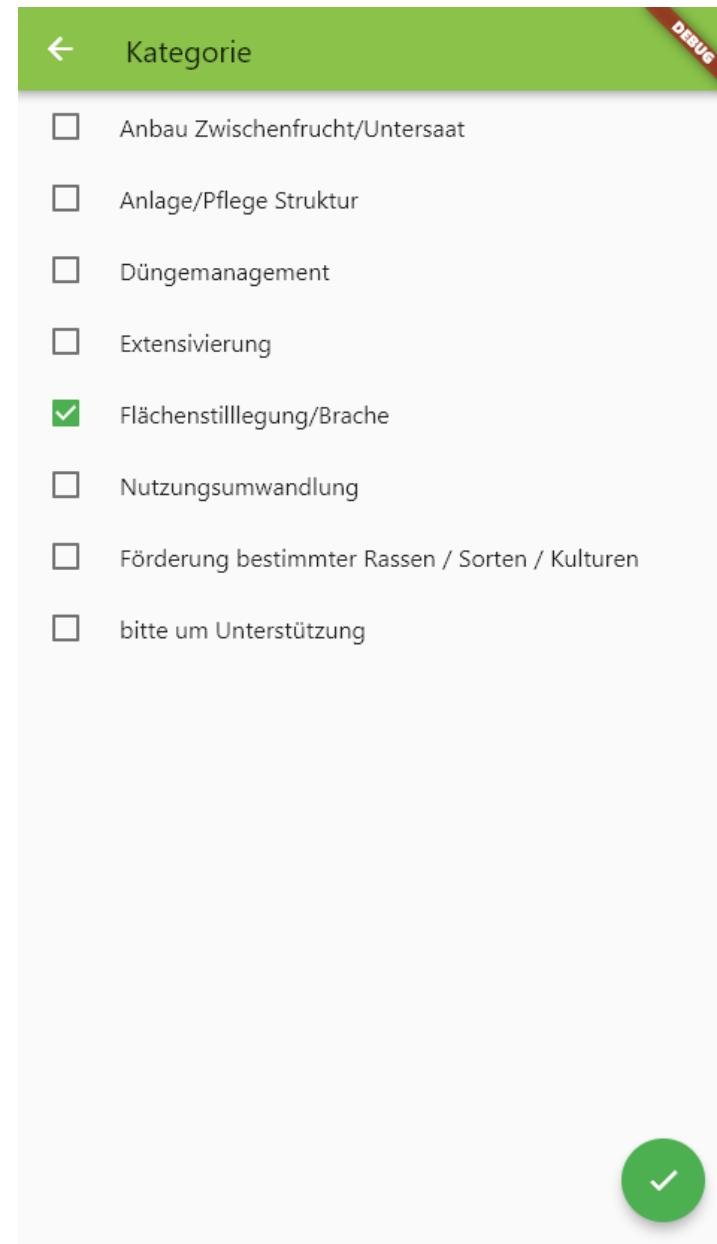
Kategorie: Flächenstilllegung/Brache (edit icon)

Zielsetzung: Zielfläche (edit icon)

Zieleinheit: Kopf/Stück (z.B. Tiere oder Bäume) (edit icon)

Hauptzielsetzung Land: (checkmark icon)

Nebenziele: (checkmark icon)



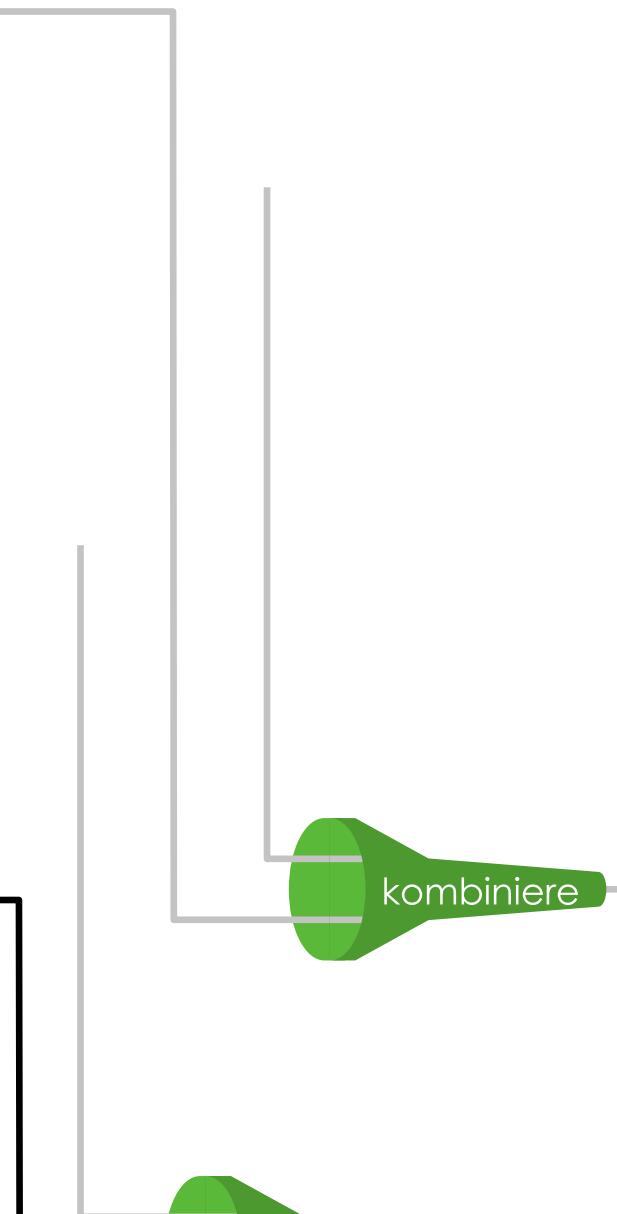
- Extensivierung
- Flächenstilllegung/Brache
- Nutzungsumwandlung
- Förderung bestimmter Rassen / Sorten / Kulturen
- bitte um Unterstützung



← Zielleinheit DEBUG

- keine Angabe/Vorgabe
- Kopf/Stück (z.B. Tiere oder Bäume)
- GV/GVE
- RGV
- ha
- bitte um Unterstützung
- m³ (z.B. Gülle)

ha



← Maßnahmen Detail DEBUG

Status
in Bearbeitung

Identifikatoren

Maßnahmentitel

Maßnahmencharakteristika

Förderklasse
Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE Vertragsnaturschutz

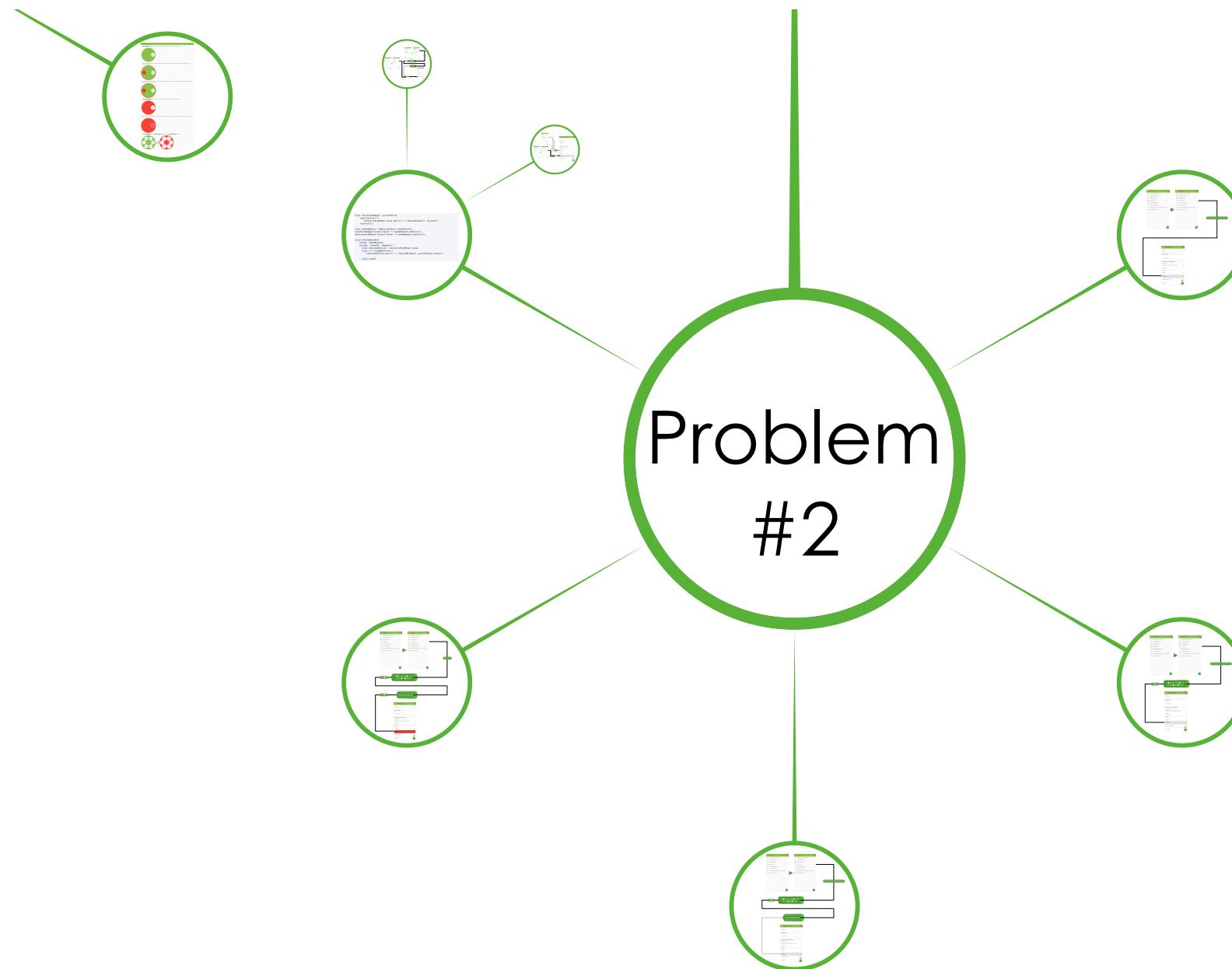
Kategorie
Flächenstilllegung/Brache

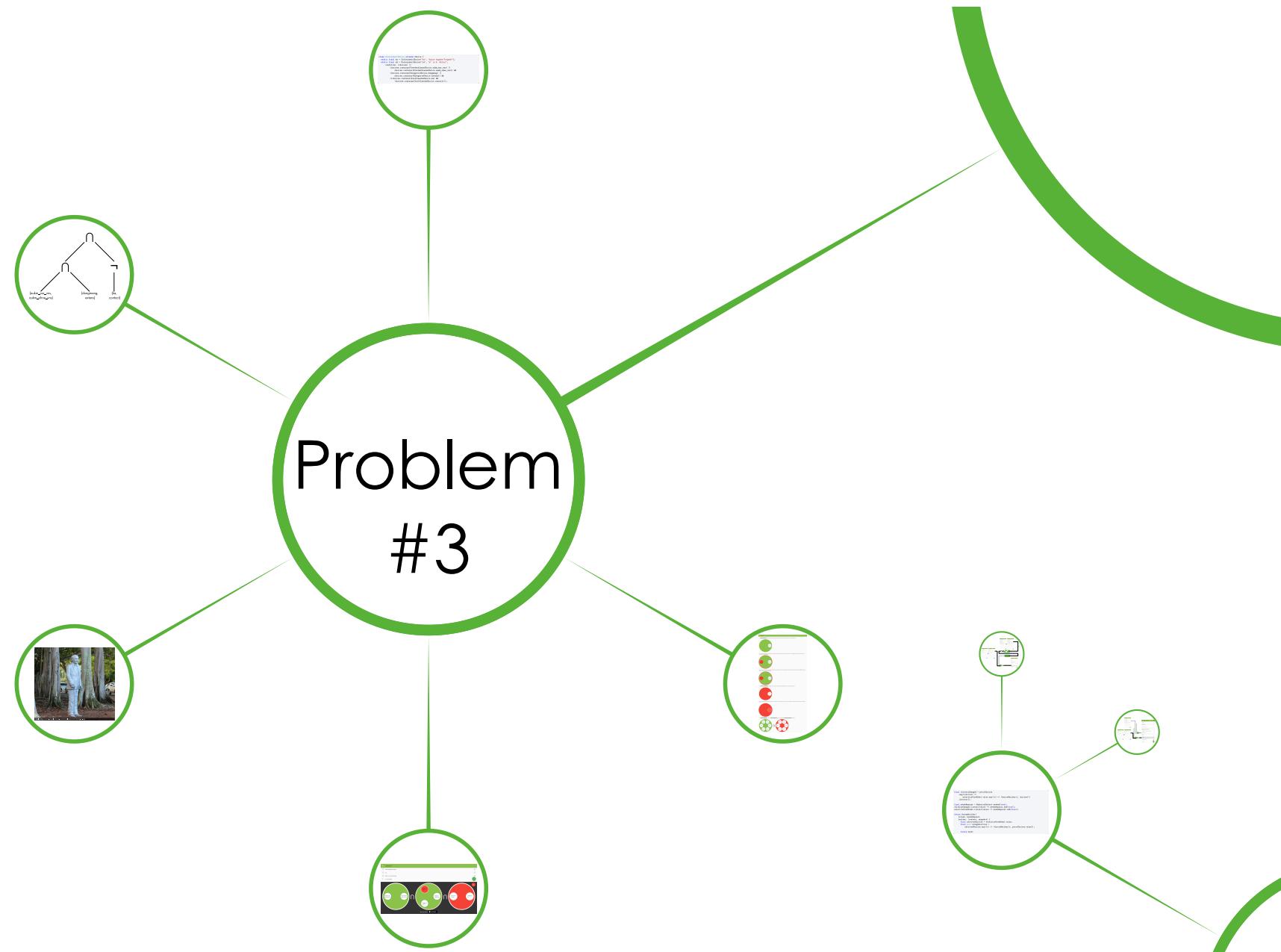
Zielsetzung

Zielfläche

Zielleinheit

Problem #2





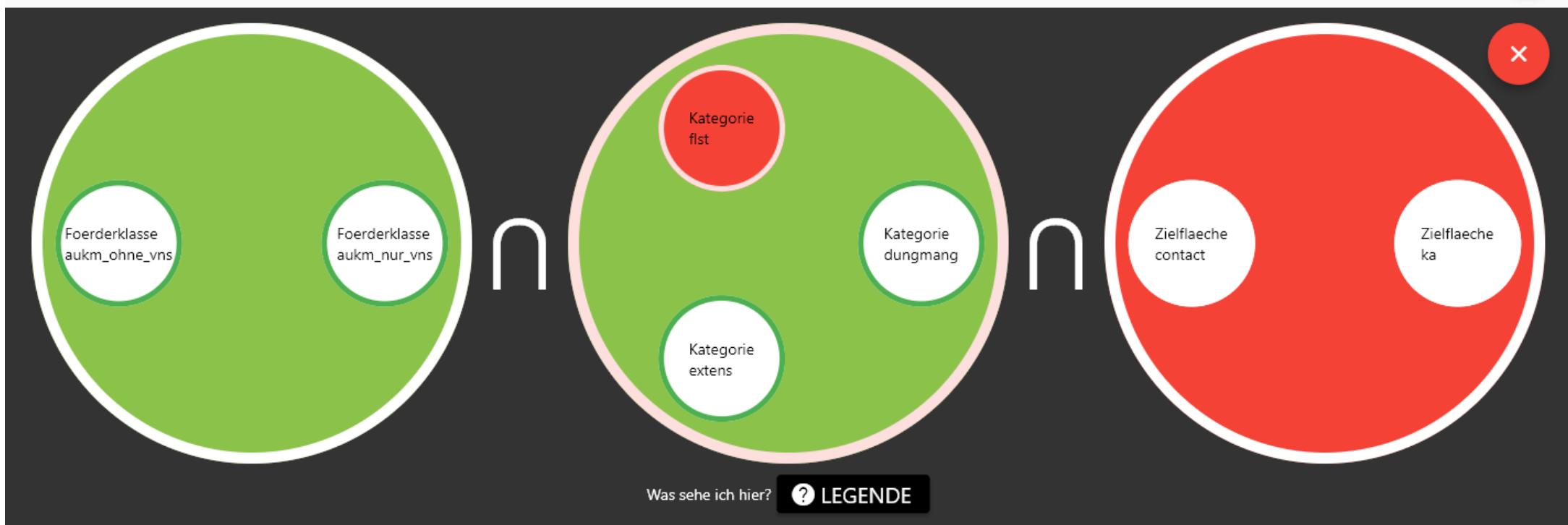
← Zieleinheit

keine Angabe/Vorgabe (i)

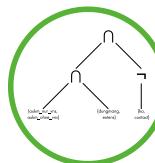
ha (i)

bitte um Unterstützung (i)

m³ (z.B. Gülle) (✓)



Problem #3



```
class ZieleinheitChoice extends Choice {  
    static final ka = ZieleinheitChoice("ka", "keine Angabe/Vorgabe");  
    static final m3 = ZieleinheitChoice("m3", "m3 (z.B. Gölle)",  
        condition: (choices) =>  
            (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||  
             choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&  
            (choices.contains(KategorieChoice.dungmang) ||  
             choices.contains(KategorieChoice.extens)) &&  
            (!choices.contains(ZielflaecheChoice.ka) &&  
             !choices.contains(ZielflaecheChoice.contact)));
```

Problem #3



```
git clone https://github.com/.../tree/main
cd main
javac Main.java
java Main
```

A screenshot of a terminal window showing Java code being run.

```
git clone https://github.com/.../tree/main
cd main
javac Main.java
java Main
```

A screenshot of a terminal window showing Java code being run.

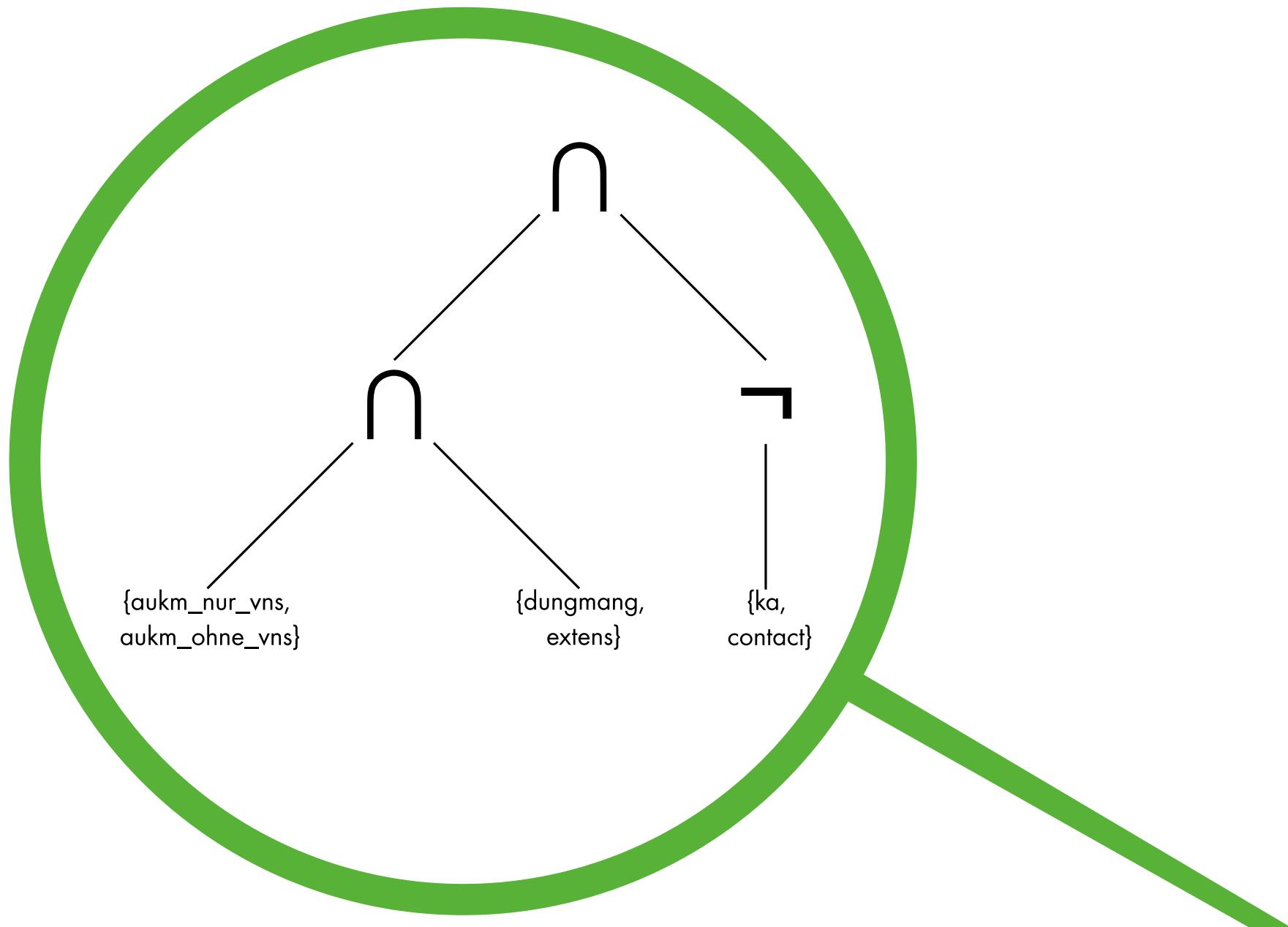


Quelle: <http://www.artswfl.com/wp-content/uploads/2012/10/006-43.jpg>





Quelle: <http://www.artswfl.com/wp-content/uploads/2012/10/006-43.jpg>



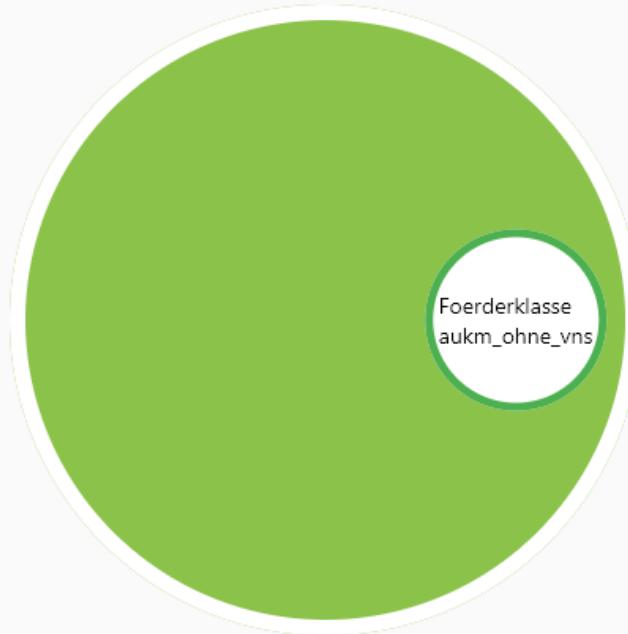


Eine **Und-Verknüpfung** zwischen einer **Muss-Beinhalten** Bedingung und einer einer **Darf-Nicht-Beinhalten** Bedingung.



← Legende

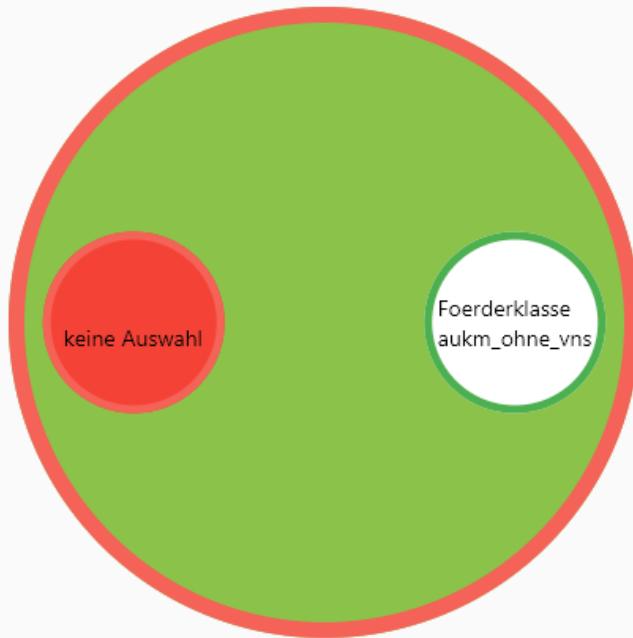
Eine **Muss-Beinhalten** Bedingung. Der grüne Mengen-Kreis signalisiert, dass die Förderklasse aukm_ohne_vns ausgewählt sein muss.



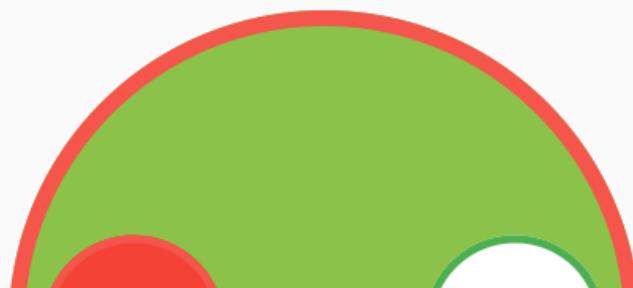
Der rot pulsierende Rahmen um den Mengenkreis signalisiert, dass die Auswahl in diesem Mengenkreis inkorrekt ist. Es ist **keine** Förderklasse ausgewählt, dabei muss aukm_ohne_vns ausgewählt sein.

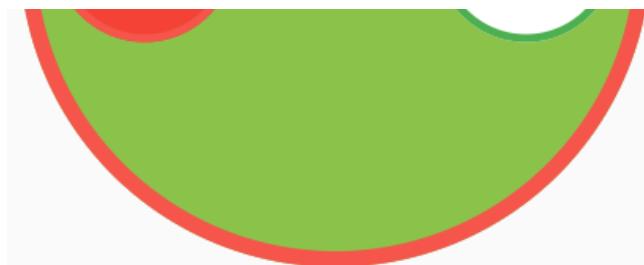


Der rot pulsierende Rahmen um den Mengenkreis signalisiert, dass die Auswahl in diesem Mengenkreis inkorrekt ist. Es ist **keine** Förderklasse ausgewählt, dabei muss aukm_ohne_vns ausgewählt sein.

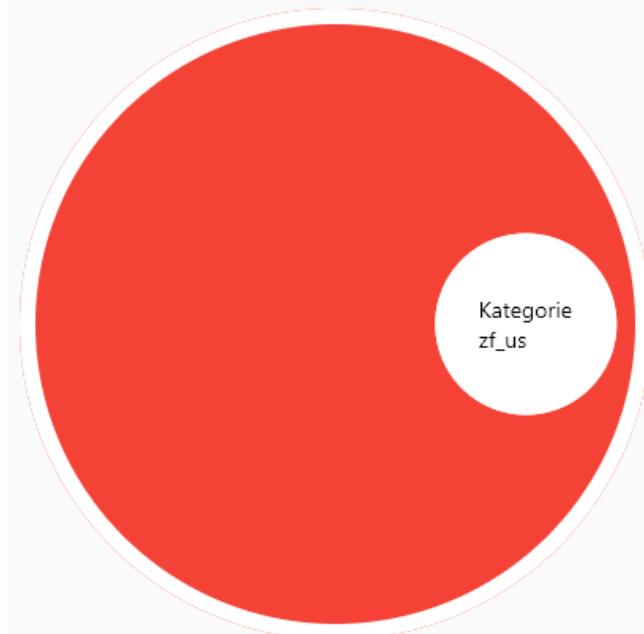


Der rot pulsierende Rahmen um die Auswahl Förderklasse ea signalisiert, dass diese Auswahl inkorrekt ist. Es ist eine Förderklasse ausgewählt, aber die **falsche** nämlich ea, dabei muss aukm_ohne_vns ausgewählt sein.





Eine **Darf-Nicht-Beinhalten** Bedingung. Der rote Mengen-Kreis signalisiert, dass die Kategorie zf_us nicht ausgewählt sein darf.

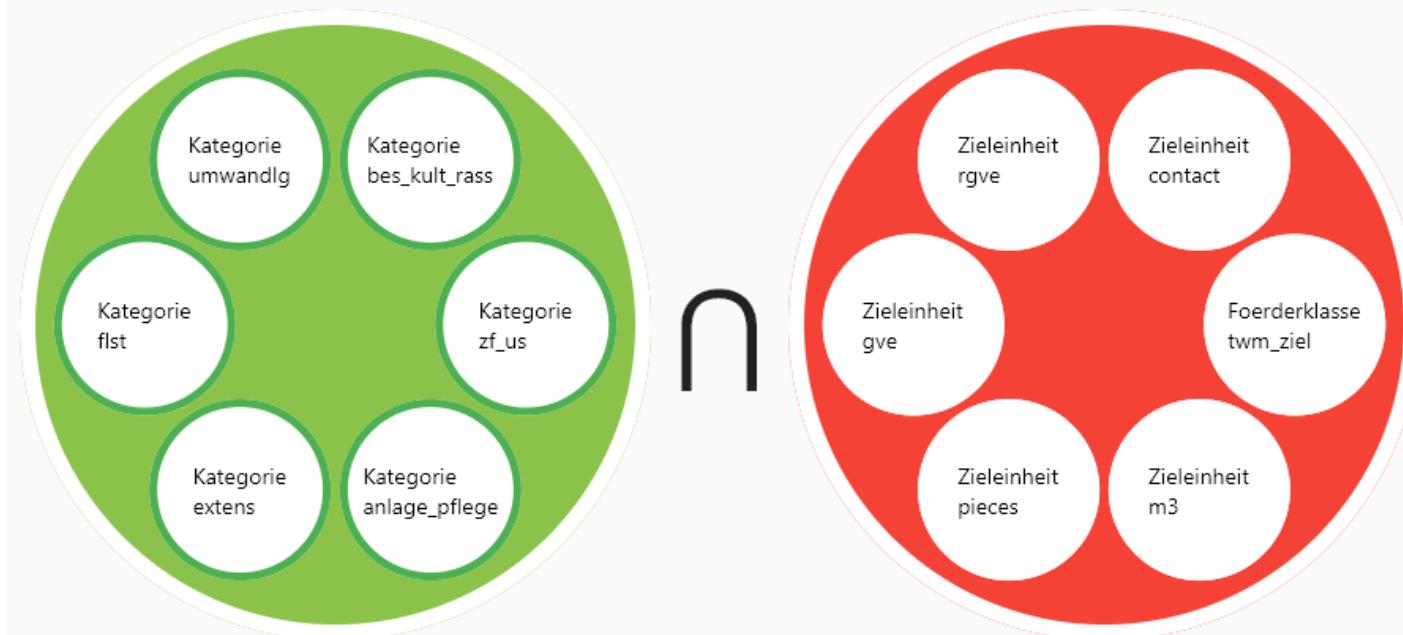


Der rot pulsierende Kreis der Auswahl Kategorie zf_us signalisiert, dass diese Auswahl inkorrekt ist. Es ist zf_us als Kategorie ausgewählt, dabei dürfte sie nicht ausgewählt sein.

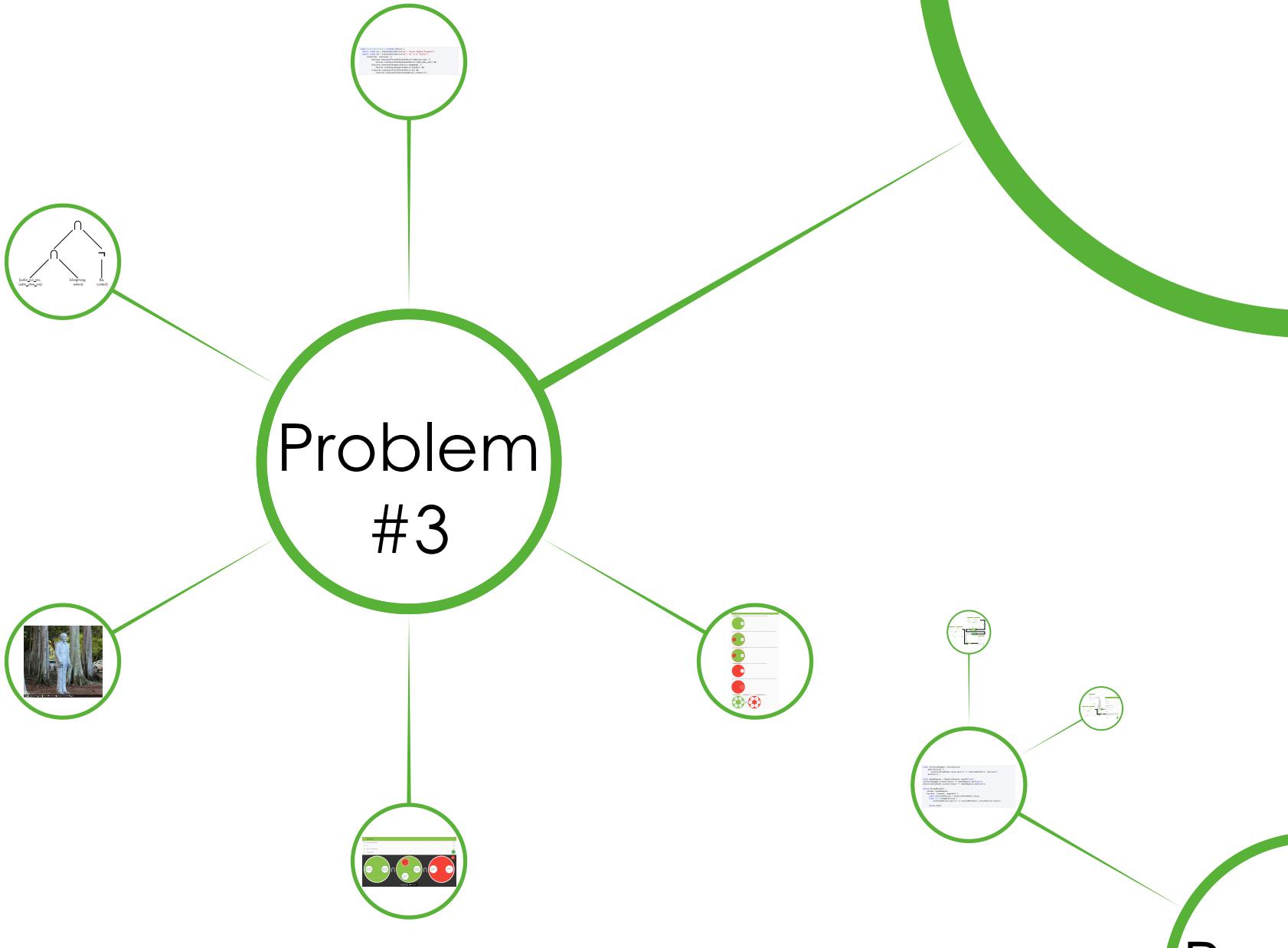


zf_us

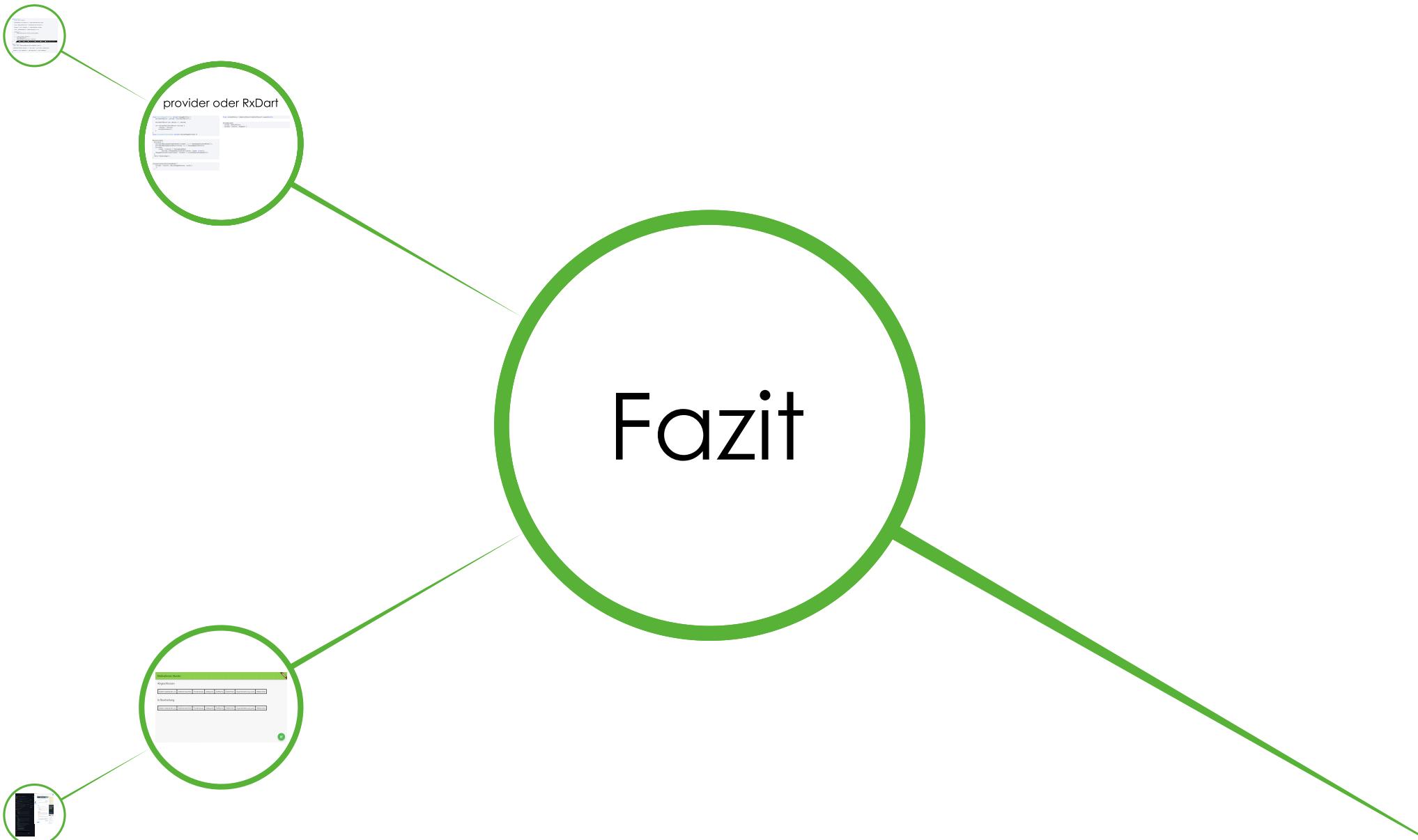
Eine **Und-Verknüpfung** zwischen einer **Muss-Beinhalten** Bedingung und einer einer **Darf-Nicht-Beinhalten** Bedingung.



Problem #3



Fazit



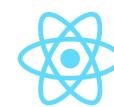
Welche Technologie ?



Test-Bibliotheken



Flutter



React Native



Flutter



Jest



React Native Testing Library



appium



Detox

Fazit



Maßnahmen Master

DEBUG

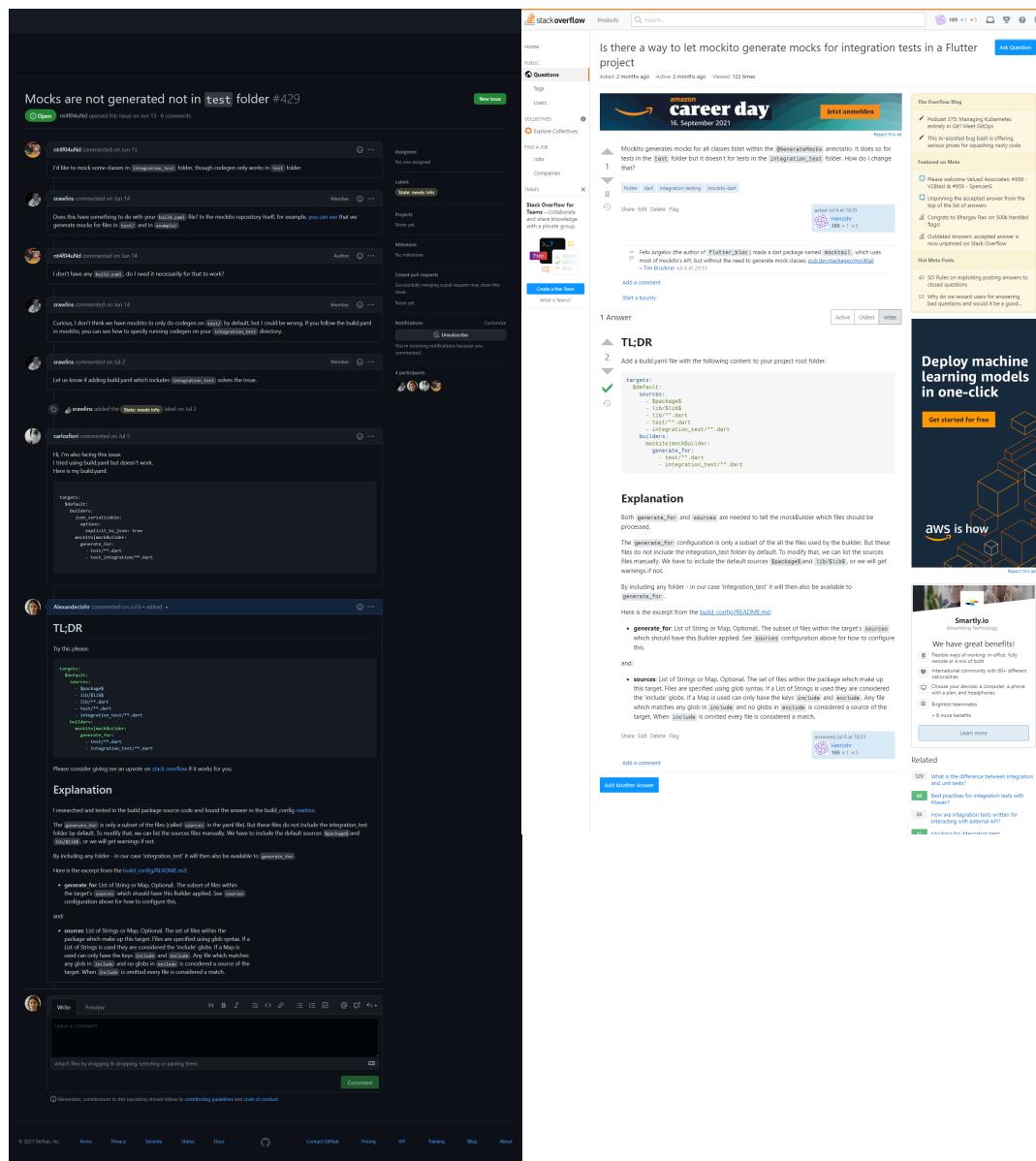
Abgeschlossen

| Zuletzt bearbeitet am | Maßnahmentitel | Förderklasse | Kategorie | Zielfläche | Zieleinheit | Hauptzielsetzung Land | Nebenziele |
|-----------------------|----------------|--------------|-----------|------------|-------------|-----------------------|------------|
|-----------------------|----------------|--------------|-----------|------------|-------------|-----------------------|------------|

In Bearbeitung

| Zuletzt bearbeitet am | Maßnahmentitel | Förderklasse | Kategorie | Zielfläche | Zieleinheit | Hauptzielsetzung Land | Nebenziele |
|-----------------------|----------------|--------------|-----------|------------|-------------|-----------------------|------------|
|-----------------------|----------------|--------------|-----------|------------|-------------|-----------------------|------------|





Mocks are not generated not in test folder #429

Open

nt4f04uNd opened this issue on Jun 13 · 6 comments



nt4f04uNd commented on Jun 13

...

I'd like to mock some classes in `integration_test` folder, though codegen only works in `test` folder



srawlins commented on Jun 14

Member

...

Does this have something to do with your `build.yaml` file? In the Mockito repository itself, for example, you can see that we generate mocks for files in `test/` and in `example/`.



nt4f04uNd commented on Jun 14

Author

...

I don't have any `build.yaml`, do I need it necessarily for that to work?



AlexanderJohr commented on Jul 6 • edited

...

TL;DR

Try this please:

```
targets:  
$default:  
sources:  
- $package$  
- lib/$lib$  
- lib/**.dart  
- test/**.dart  
- integration_test/**.dart  
builders:  
mockito|mockBuilder:  
generate_for:  
- test/**.dart  
- integration_test/**.dart
```

Please consider giving me an upvote on [stack overflow](#) if it works for you.

Explanation

Explanation

I researched and tested in the build package source code and found the answer in the `build_config` `readme`.

The `generate_for` is only a subset of the files (called `sources` in the yaml file). But these files do not include the `integration_test` folder by default. To modify that, we can list the sources files manually. We have to include the default sources `$package$` and `lib/lib`, or we will get warnings if not.

By including any folder - in our case '`integration_test`' it will then also be available to `generate_for`.

Here is the excerpt from the `build_config/README.md`:

- **generate_for:** List of String or Map, Optional:. The subset of files within the target's `sources` which should have this Builder applied. See `sources` configuration above for how to configure this.

and:

- **sources:** List of Strings or Map, Optional. The set of files within the package which make up this target. Files are specified using glob syntax. If a List of Strings is used they are considered the 'include' globs. If a Map is used can only have the keys `include` and `exclude`. Any file which matches any glob in `include` and no globs in `exclude` is considered a source of the target. When `include` is omitted every file is considered a match.

stack overflow Products Search... 189 • 1 • 5 Ask Question

Home PUBLIC Questions Tags Users COLLECTIVES Explore Collectives FIND A JOB Jobs Companies TEAMS Stack Overflow for Teams – Collaborate and share knowledge with a private group. Create a free Team What is Teams?

Is there a way to let Mockito generate mocks for integration tests in a Flutter project

Asked 2 months ago Active 2 months ago Viewed 122 times

 Report this ad

Mockito generates mocks for all classes listed within the `@GenerateMocks` annotation. It does so for tests in the `test` folder but it doesn't for tests in the `integration_test` folder. How do I change that?

1 flutter dart integration-testing Mockito-Dart Share Edit Delete Flag asked Jul 6 at 18:35 by HerrJohr 189 • 1 • 5

Felix Angelov (the author of `flutter_bloc`) made a Dart package named `mocktail`, which uses most of Mockito's API, but without the need to generate mock classes. pub.dev/packages/mocktail

– Tim Brückner Jul 6 at 20:13

Add a comment Start a bounty

The Overflow Blog

- Podcast 375: Managing Kubernetes entirely in Git? Meet GitOps
- This AI-assisted bug bash is offering serious prizes for squashing nasty code

Featured on Meta

- Please welcome Valued Associates: #958 - V2Blast & #959 - SpencerG
- Unpinning the accepted answer from the top of the list of answers
- Congrats to Bhargav Rao on 500k handled flags!
- Outdated Answers: accepted answer is now unpinned on Stack Overflow

Hot Meta Posts

- 40 SO Rules on exploiting posting answers to closed questions
- 32 Why do we reward users for answering

TL;DR

2

Add a build.yaml file with the following content to your project root folder.

```
targets:  
  $default:  
    sources:  
      - $package$  
      - lib/$lib$  
      - lib/**.dart  
      - test/**.dart  
      - integration_test/**.dart  
    builders:  
      mockito|mockBuilder:  
        generate_for:  
          - test/**.dart  
          - integration_test/**.dart
```

Explanation

Both `generate_for` and `sources` are needed to tell the `mockBuilder` which files should be processed.

The `generate_for` configuration is only a subset of the all the files used by the builder. But these files do not include the `integration_test` folder by default. To modify that, we can list the `sources` files manually. We have to include the default sources `$package$` and `lib/lib`, or we will get warnings if not.

By including any folder - in our case '`integration_test`' it will then also be available to `generate_for`.

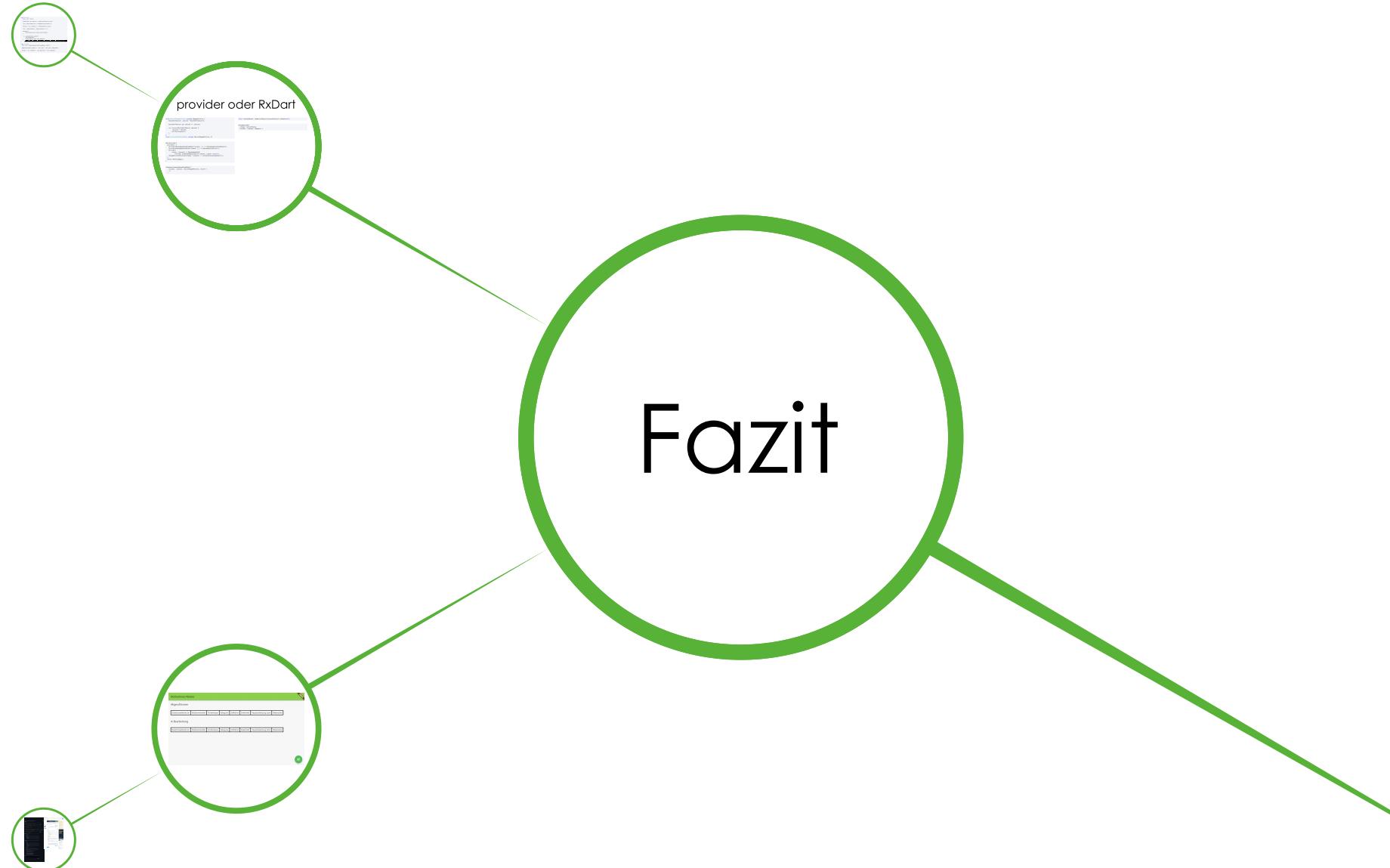
[Here is the excerpt from the `build_config/README.md`.](#)

Deploy machine learning models in one-click

[Get started for free](#)

aws is how

[Report this slide](#)

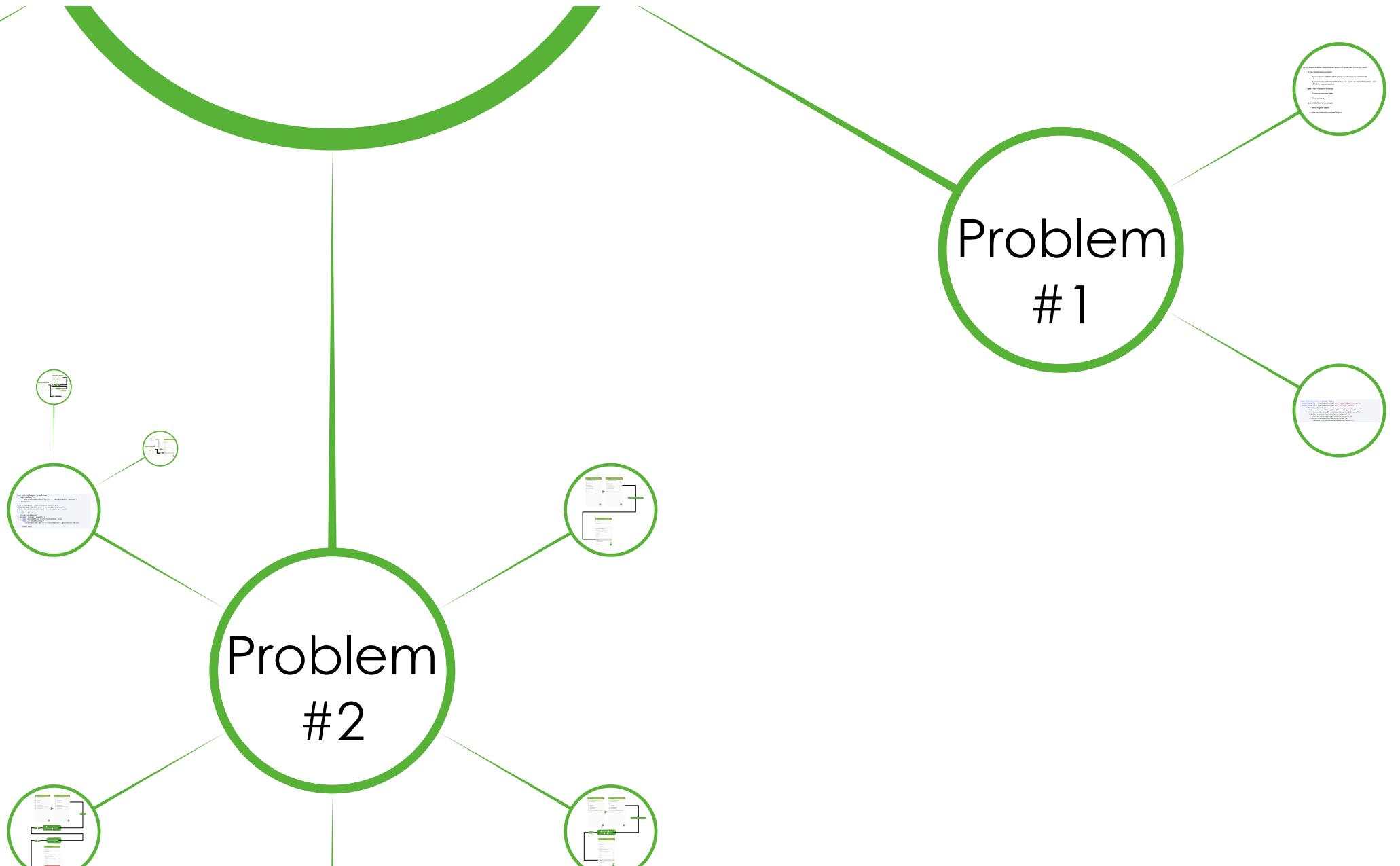


Problem # 1

Um im Auswahlfeld der Zeileheit die Option **mit** auszuwählen zu können müssen Sie die Zeile mit dem Mauszeiger überfahren.

- für die Pfeilhäuser entweder
 - Agrarwelt und Kleinbaustruktur: nur Wettbewerbsnachteil oder
 - Agrarwelt und Kleinstadtstruktur, zw. auch mit Wettbewerbsvorteilen, alle
 - und für die Regionen entweder
 - Deregulierung oder
 - Erweiterung
 - und für Ziele von **weder**
 - keine Angabe, noch
 - bitte um Unterstützung gewünscht

```
class ZieleinheitChoice extends Choice {  
    static final ka = ZieleinheitChoice("ka", "keine Angabe/Vorgabe");  
    static final m3 = ZieleinheitChoice("m3", "m3 (z.B. Gölle)",  
        condition: (choices) =>  
            (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||  
             choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&  
            (choices.contains(KategorieChoice.dungmang) ||  
             choices.contains(KategorieChoice.extens)) &&  
            (!choices.contains(ZielflaecheChoice.ka) &&  
             !choices.contains(ZielflaecheChoice.contact)));
```

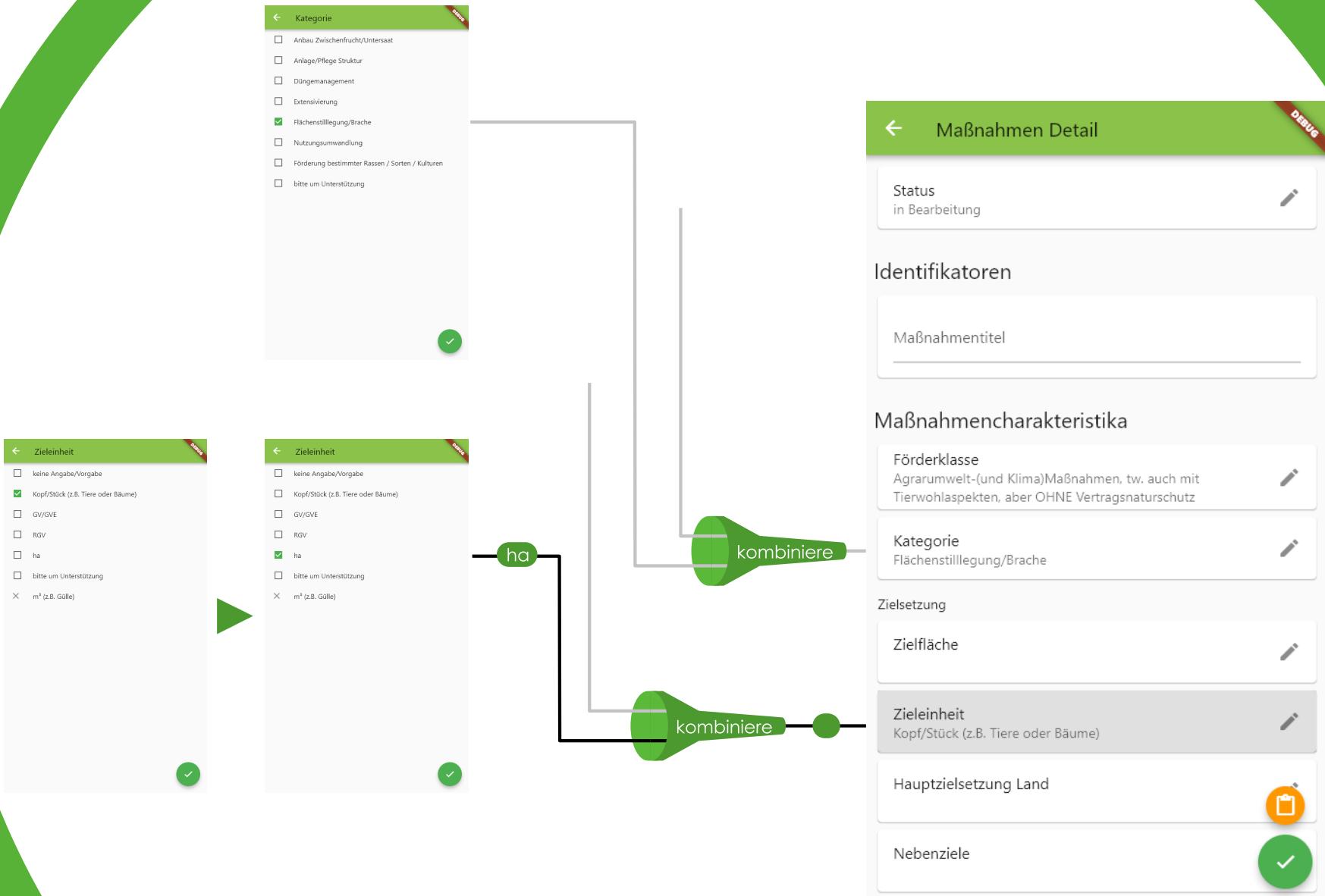


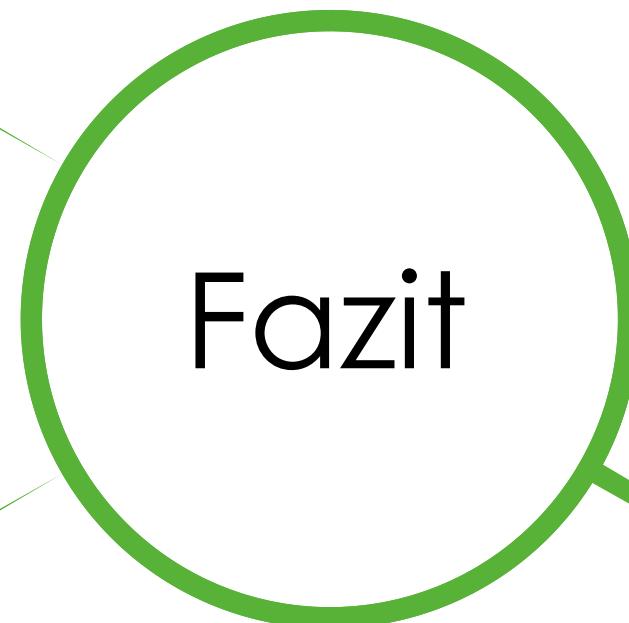
```
final validityChanged = priorChoices
    .map((choices) =>
        selectionViewModel.value.any((c) => !choiceMatcher(c, choices)))
    .distinct();

final needsRepaint = BehaviorSubject.seeded(true);
validityChanged.listen((value) => needsRepaint.add(true));
selectionViewModel.listen((value) => needsRepaint.add(true));

return StreamBuilder(
    stream: needsRepaint,
    builder: (context, snapshot) {
        final selectedChoices = selectionViewModel.value;
        final bool wrongSelection =
            selectedChoices.any((c) => !choiceMatcher(c, priorChoices.value));

        return Card(
```





provider oder RxDart

```
class ChoiceChangeNotifier extends ChangeNotifier {
  BuiltSet<Choice> _choices = BuiltSet<Choice>();

  BuiltSet<Choice> get choices => _choices;

  set choices(BuiltSet<Choice> choices) {
    _choices = choices;
    notifyListeners();
  }
}

class LetzterStatusViewModel extends ChoiceChangeNotifier {}
```

```
MultiProvider(
  providers: [
    Provider<MassnahmenFormViewModel>(create: (_) => MassnahmenFormViewModel()),
    Provider<MassnahmenJsonFile>(create: (_) => MassnahmenJsonFile()),
    Provider(
      create: (context) => MassnahmenModel(
        Provider.of<MassnahmenJsonFile>(context, listen: false))),
    ChangeNotifierProvider(create: (context) => LetzterStatusViewModel()),
  ],
  child: MaterialApp(),
)
```

```
Consumer<LetzterStatusViewModel>(
  builder: (context, choiceChangeNotifier, child) {
  },
)
```

```
final letzterStatus = BehaviorSubject<LetzterStatus?>.seeded(null);

StreamBuilder(
  stream: letzterStatus,
  builder: (context, snapshot) {
```

```
class ChoiceChangeNotifier extends ChangeNotifier {  
    BuiltSet<Choice> _choices = BuiltSet<Choice>();  
  
    BuiltSet<Choice> get choices => _choices;  
  
    set choices(BuiltSet<Choice> choices) {  
        _choices = choices;  
        notifyListeners();  
    }  
}  
class LetzterStatusViewModel extends ChoiceChangeNotifier {}
```

```
        }
    }
class LetzterStatusViewModel extends ChoiceChangeNotifier {}
```

```
MultiProvider(
    providers: [
        Provider<MassnahmenFormViewModel>(create: (_) => MassnahmenFormViewModel()),
        Provider<MassnahmenJsonFile>(create: (_) => MassnahmenJsonFile()),
        Provider(
            create: (context) => MassnahmenModel(
                Provider.of<MassnahmenJsonFile>(context, listen: false)),
            ChangeNotifierProvider(create: (context) => LetzterStatusViewModel())),
    ],
    child: MaterialApp(),
)
```

```
Consumer<LetzterStatusViewModel>(
    builder: (context, choiceChangeNotifier, child) {
```

```
」,  
child: MaterialApp(),  
)
```

```
Consumer<LetzterStatusViewModel>(  
    builder: (context, choiceChangeNotifier, child) {  
    },  
)
```

```
final letzterStatus = BehaviorSubject<LetzterStatus?>.seeded(null);
```

```
StreamBuilder(  
    stream: letzterStatus,  
    builder: (context, snapshot) {
```

provider oder RxDart

```

class ChoiceChangeNotifier extends ChangeNotifier {
  BuiltSet<Choice> _choices = BuiltSet<Choice>();
  StreamBuilder<
    Stream<LetzterStatus>,
    StreamBuilder<LetzterStatus, void>
  > builder;
}

class LetzterStatusViewModel extends ChoiceChangeNotifier {}

MultiProvider(
  providers: [
    Provider<MassnahmenFormViewModel>(create: () => MassnahmenFormViewModel()),
    Provider<MassnahmenJsonFile>(create: () => MassnahmenJsonFile()),
    Provider<ChangeNotifier>(create: (context) => MassnahmenModel(),
      create: (context) => MassnahmenModel(),
      Provider.of<MassnahmenJsonFile>(context, listen: false)),
    ChangeNotifierProvider(create: (context) => LetzterStatusViewModel()),
  ],
  child: MaterialApp(),
)

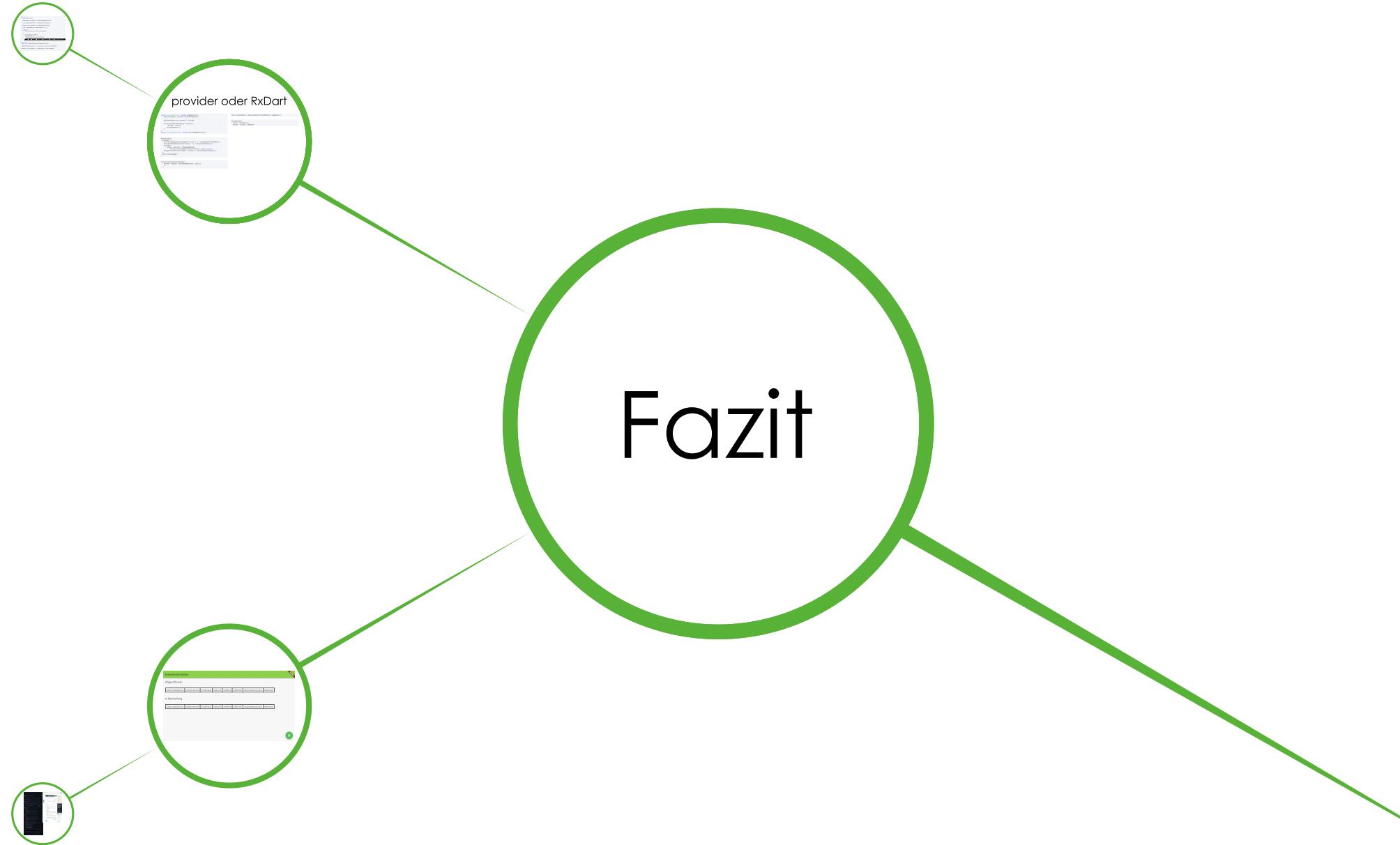
Consumer<LetzterStatusViewModel>(
  builder: (context, choiceChangeNotifier, child) {
    },
),

```

```
class CartBloc{  
    final _cart = Cart();  
  
    Sink<Product> get addition => _additionController.sink;  
  
    final _additionController = StreamController<Product>();  
  
    Stream<int> get itemCount => _itemCountSubject.stream;  
  
    final _itemCountSubject = BehaviorSubject<int>();  
  
    CartBloc(){  
        _additionController.stream.listen(_handle);  
    }  
  
    void _handle(Product product){  
        _cart.add(product);  
        _itemCountSubject.add(_cart.itemCount);  
    }  
}
```

Quelle: Google LLC, Build reactive mobile apps with Flutter (Google I/O '18) TC: 27:37

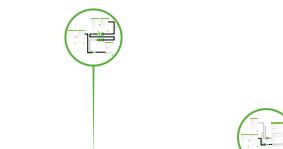
```
class CartBloc{  
    final _cart = BehaviorSubject<Cart>(seedValue: Cart());  
  
    addProduct(Product product) => _cart.value = _cart.value..add(product);  
  
    Stream<int> get itemCount => _cart.map((cart) => cart.itemCount);  
}
```



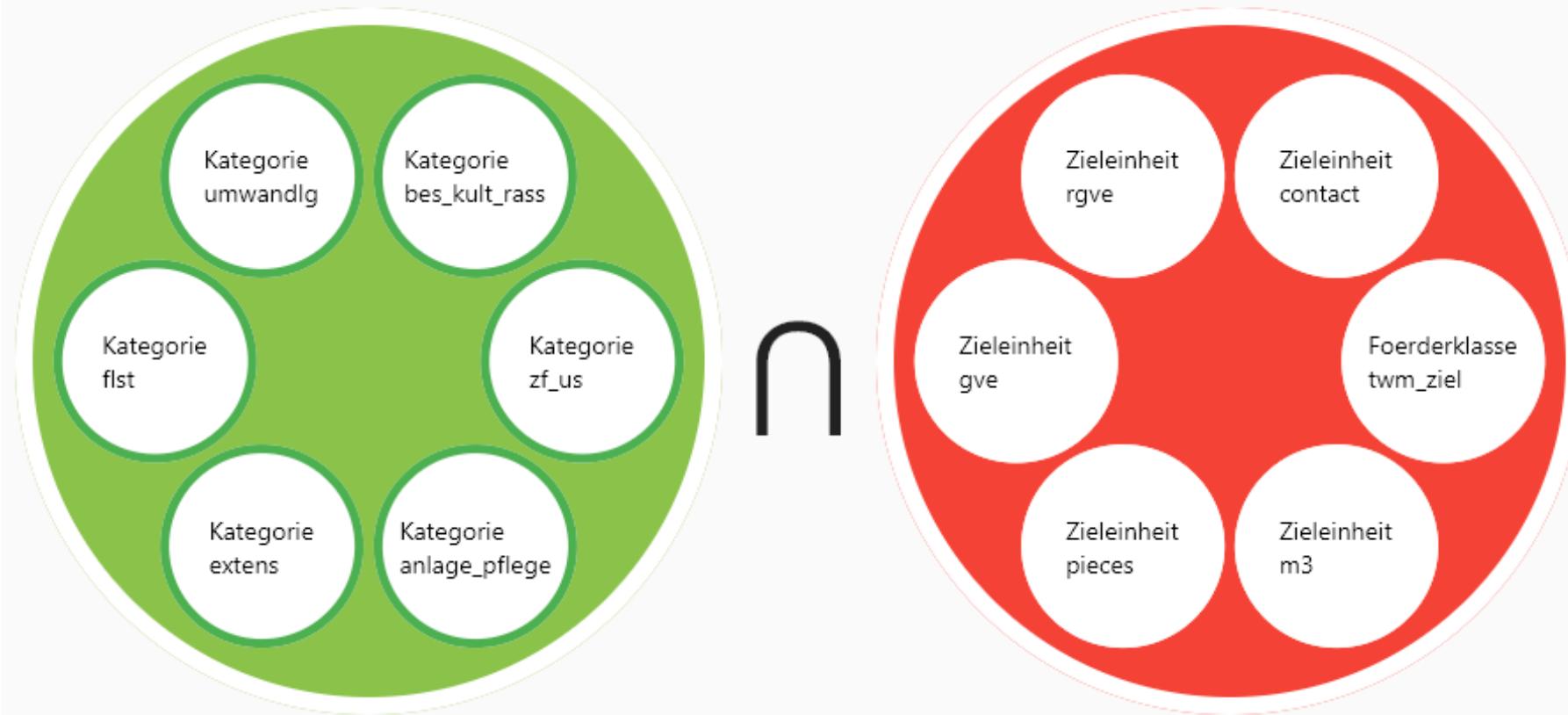
Problem #3



LINKED LIST IMPLEMENTATION



Eine **Und-Verknüpfung** zwischen einer **Muss-Beihalten** Bedingung und einer einer **Darf-Nicht-Beihalten** Bedingung.





Entwicklung einer Formularanwendung
mit Kompatibilitätsvalidierung der
Einfach- und Mehrfachauswahl-Eingabefelder

