ssystems

Umfassende, Professionelle IT-Serviceleistungen

Firmengeschichte

Gegründet im Jahr 2003 in Umfeld der Hochschule München verfügt ssystems über mehr als 10 Jahre Erfahrung in der Zusammenarbeit mit Personen, Gruppen, Teams in Hochschulen, die sich mit Campus-IT beschäftigen.

ssystems mit seinen Sitzen in München und Berlin ist derzeit eines der wenigen Unternehmen in Deutschland, die mit den Herausforderungen von Campus-IT auch im Kontext großer Hochschulen vertraut sind und bei dem das konzeptionelle und technische Fachwissen existiert, um ihre Projekt kompetent begleiten zu können.

Wir sind groß genug, um komplexe Lösungen realisieren zu können und dabei menschlich präsent und organisatorisch flexibel.

Über uns

Umfassende, Professionelle IT-Serviceleistungen

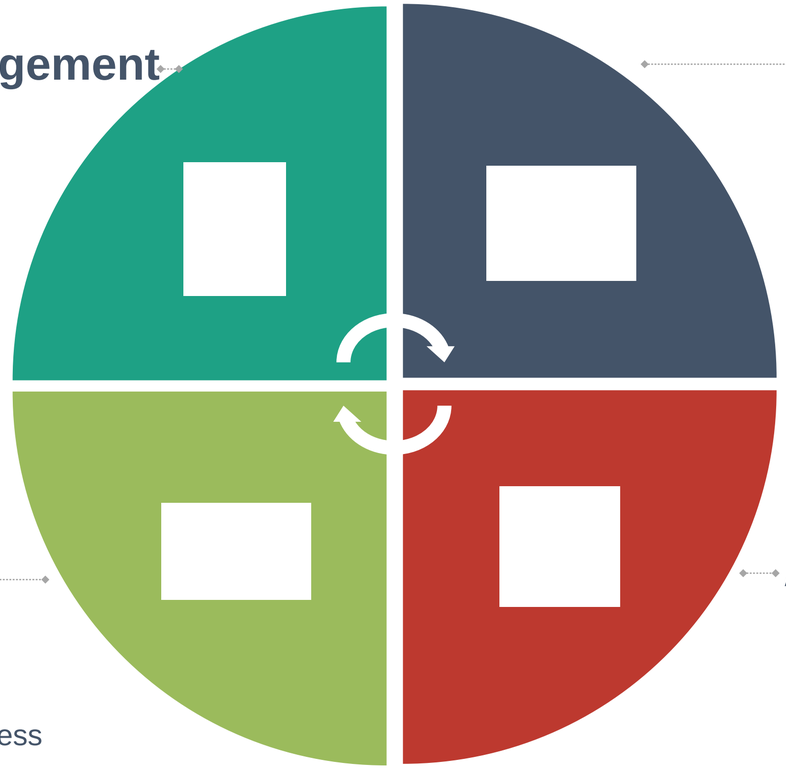
CAMPUSSPEICHER
Hosting mit Geist.

Identity und Access Management

Beratung, Konzeption und Umsetzung
Verwaltung und Betrieb
Metadirectories und Prozesberatung
Verzeichnisdienste
SSO, Shibboleth und AAI
Erweiterung und Integration

Server und Infrastruktur

Campus Management
Mail, Web, Storage
(Managed) Hosting, IaaS
UNIX, Linux und Netzwerke
Virtualisierung
Datenschutz



E-Learning, E-Prüfung

Moodle, Blackboard, CLIX
Social Learning, E-Portfolios, Mahara
Online Klausuren – E-Assessment
CMS: TYPO3, Drupal, Firstspirit, Wordpress

Anwendungsentwicklung

CMS und LMS Erweiterungen
Systemprogrammierung, Backends
Customized Frontends
JAVA, Spring, GWT, BPM
Perl, PHP, Shell, uvm.

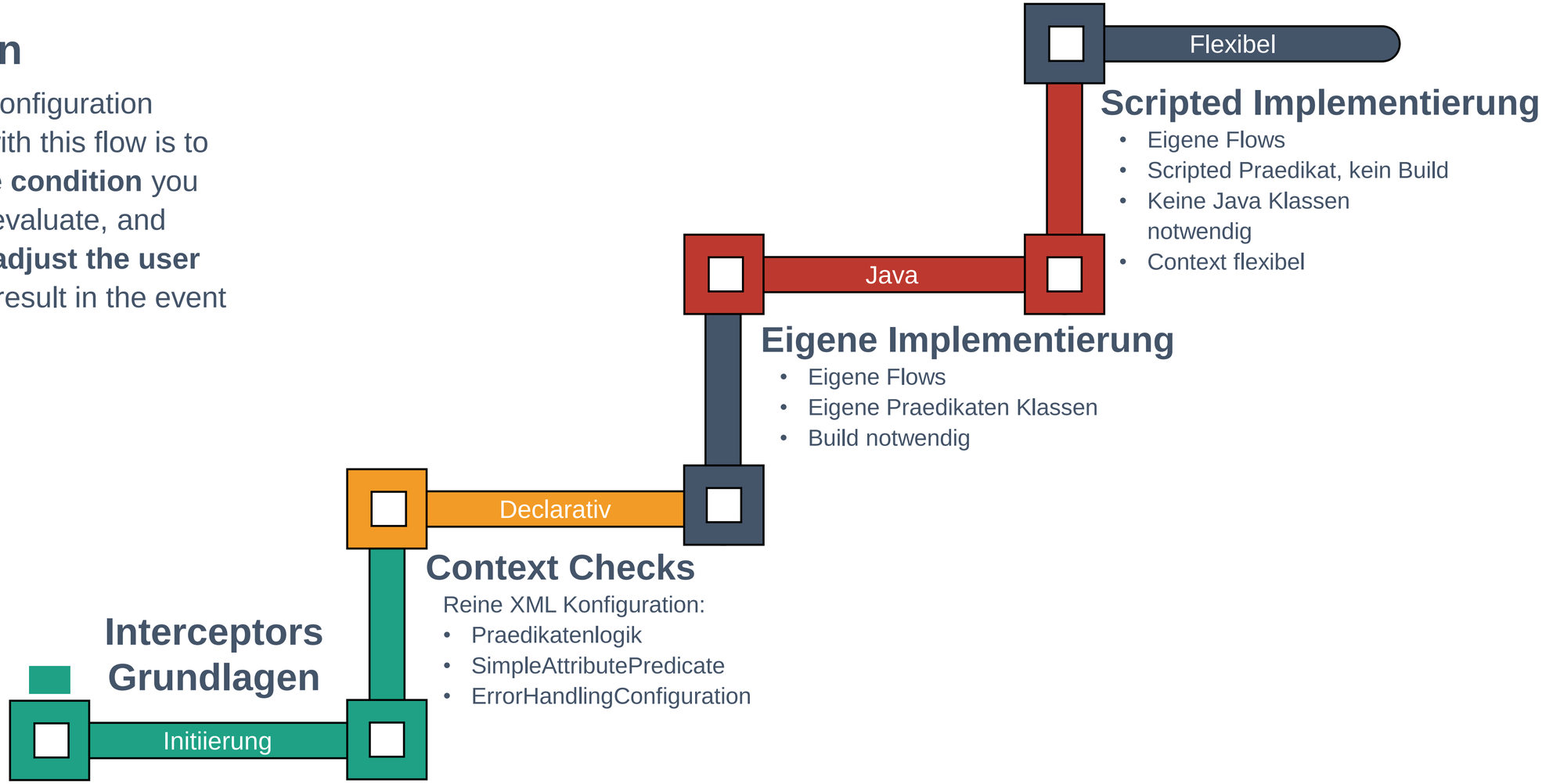
undkonsorten

Interceptors

Typen

Grundlagen

- The only configuration involved with this flow is to **define the condition** you want it to evaluate, and **possibly adjust the user interface** result in the event of failure.

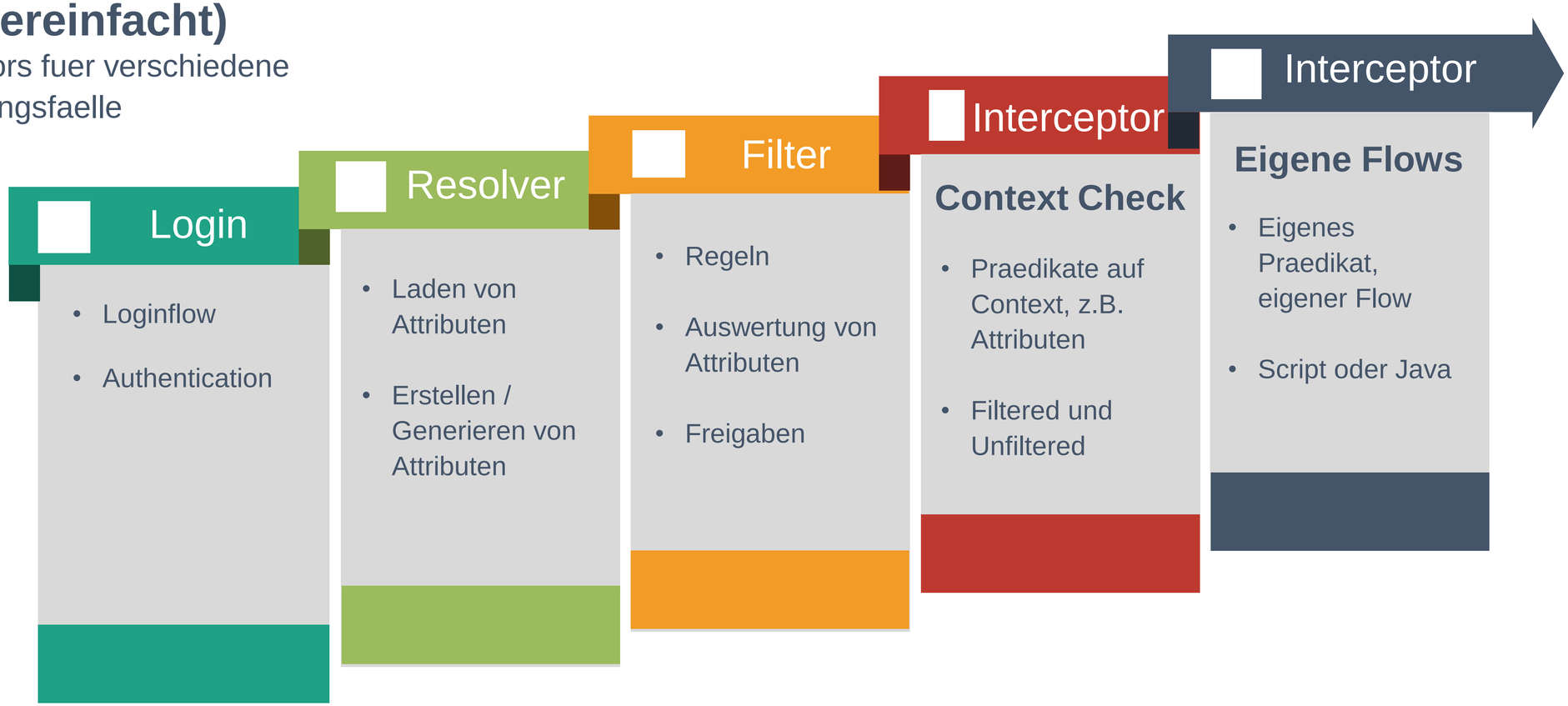


Interceptors

Was will man erreichen?

Flow (vereinfacht)

Interceptors fuer verschiedene Anwendungsfaelle

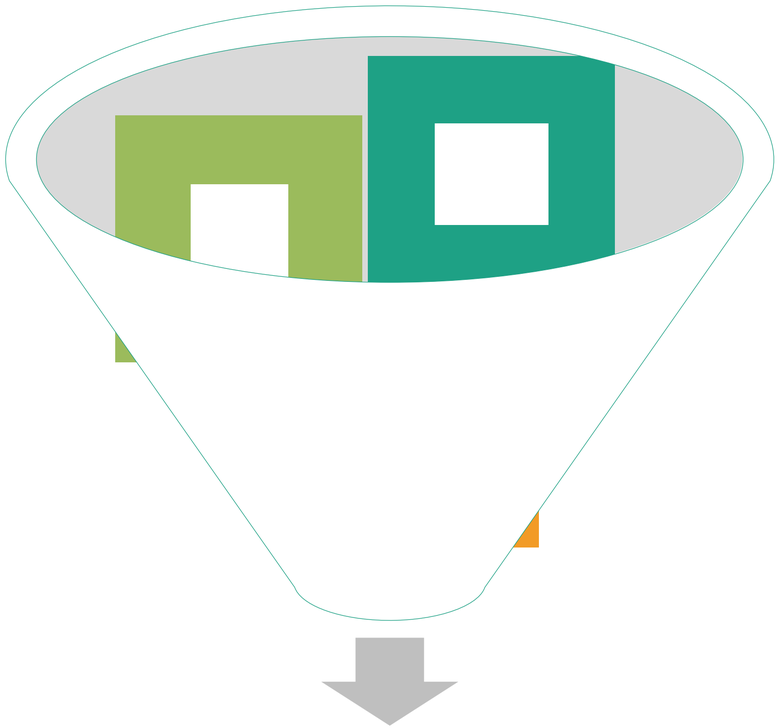


Context Check

Nicht jeden in die DFN-AAI lassen

Deklarativ

- Einfach, oft ausreichend



consent-intercept-config.xml

(Attributbezogene) Bedingungen, liefern true / false zurueck

relaying-party.xml

Interceptor fuer Profile aktivieren (Shib1, Shib2)

Template

Anzeige von Meldung. Default: Access Denied

Profile Configuration

vi relaying-party.xml

```
<bean id="shibboleth.DefaultRelyingParty"
parent="RelyingParty">
 <property name="profileConfigurations">
    <list>
     <bean parent="Shibboleth.SSO"
p:postAuthenticationFlows="#{{'attribut
e-release', 'context-check'}}"/>
     <ref bean="SAML1.AttributeQuery" />
     <ref bean="SAML1.ArtifactResolution" />
     <ref bean="SAML2.SSO.custom" />
     <ref bean="SAML2.ECP" />
     <ref bean="SAML2.Logout" />
     <ref bean="SAML2.AttributeQuery" />
     <ref bean="SAML2.ArtifactResolution" /> <ref
bean="Liberty.SSOS" />
    </list>
 </property>
 </bean>
```

Predicate Configuration

vi conf/intercept/context-check-intercept-config.xml

DFN Policy umsetzen

- Nur identifizierte Personen in die DFN-AAI lassen
- isIdentified wird aus LDAP Attribut (Accounteigenschaft) in attribute-resolver.xml generiert

```
<bean id="shibboleth.context-check.Condition"
parent="shibboleth.Conditions.OR">
   <constructor-arg>
      <list>
         <bean parent="shibboleth.Conditions.NOT">
            <constructor-arg>
               <list>
                  <bean
class="net.shibboleth.idp.profile.logic.SimpleAttributePredicate">
                     <property name="attributeValueMap">
                        <map>
                           <entry key="isIdentified">
                              <list>
                                 <value>false</value>
                              </list>
                           </entry>
                        </map>
                     </property>
                  </bean>
               </list>
            </constructor-arg>
         </bean>
```

Predicate Configuration

vi conf/intercept/context-check-intercept-config.xml

Lokale Policy umsetzen

- Ausnahmen fuer interne SPs setzen

```
<bean parent="shibboleth.Conditions.AND">
    <constructor-arg>
        <list>
            <bean class="net.shibboleth.idp.profile.logic.SimpleAttributePredicate">
                <property name="attributeValueMap">
                    <map>
                        <entry key="isIdentified">
                            <list>
                                <value>false</value>
                            </list>
                        </entry>
                    </map>
                </property>
            </bean>
            <bean parent="shibboleth.Conditions.RelyingPartyId">
                <constructor-arg>
                    <list>
                        <value>https://sp1.example.edu/shibboleth</value>
                        <value>https://sp2.example.edu/shibboleth</value>
                        <value>https://sp3.example.edu/shibboleth</value>
                    </list>
                </constructor-arg>
            </bean>
        </list>
    </constructor-arg>
</bean>
```

Predicate Configuration

vi conf/intercept/context-check-intercept-config.xml

Lokale Policy umsetzen

- Lokale SPs als Properties konfigurieren

```
<bean parent="shibboleth.Conditions.AND">
        <constructor-arg>
          <list>
            <bean class="net.shibboleth.idp.profile.logic.SimpleAttributePredicate">
              <property name="attributeValueMap">
                <map>
.....

<bean parent="shibboleth.Conditions.RelyingPartyId">
  <constructor-arg name="candidates">
    <bean parent="shibboleth.CommaDelimStringArray" c:_0="#{ '%{idp.local.services}'.trim() }" />
  </constructor-arg>
</bean>
                </list>
              </constructor-arg>
            </bean>
          </list>
        </constructor-arg>
</bean>
```

idp.local.services = https://sp1.example.org, https://sp2.example.org

Predicate Interface

Sehr schmal

Definition

- Google collections library
- Wird an vielen Stellen in IdP 3 verwendet
- Muss true|false zurueckliefern
- Sonst keine Grenzen

```
public interface Predicate<T> {
  /**
   * Returns the result of applying this predicate to {@code input}. This method
is <i>generally
   * expected</i>, but not absolutely required, to have the following properties:
   *
   * <ul>
   * <li>Its execution does not cause any observable side effects.
   * <li>The computation is <i>consistent with equals</i>; that is, {@link
Objects#equal
   *     Objects.equal}{@code (a, b)} implies that {@code predicate.apply(a) ==
   *     predicate.apply(b))}.
   * </ul>
   *
   * @throws NullPointerException if {@code input} is null and this predicate
does not accept null
   *     arguments
   */
  boolean apply(@Nullable T input);
```

Predicate Types

Sehr, sehr viele

Implementierungen

- Google collections library
- Shibboleth
- OpenSAML

public class AbstractAttributePredicate …. (filtered, unfiltered)

public class DateAttributePredicate…. (now() > any attribute value)

public class IPRangePredicate ….

public class ScriptedPredicate ….

uvm.

Praedicate Class

vi …./AnyAttributePredicate.java

Erweiterung

- AbstractAttributePredicate erweitern
- Sind Attribute freigegeben?

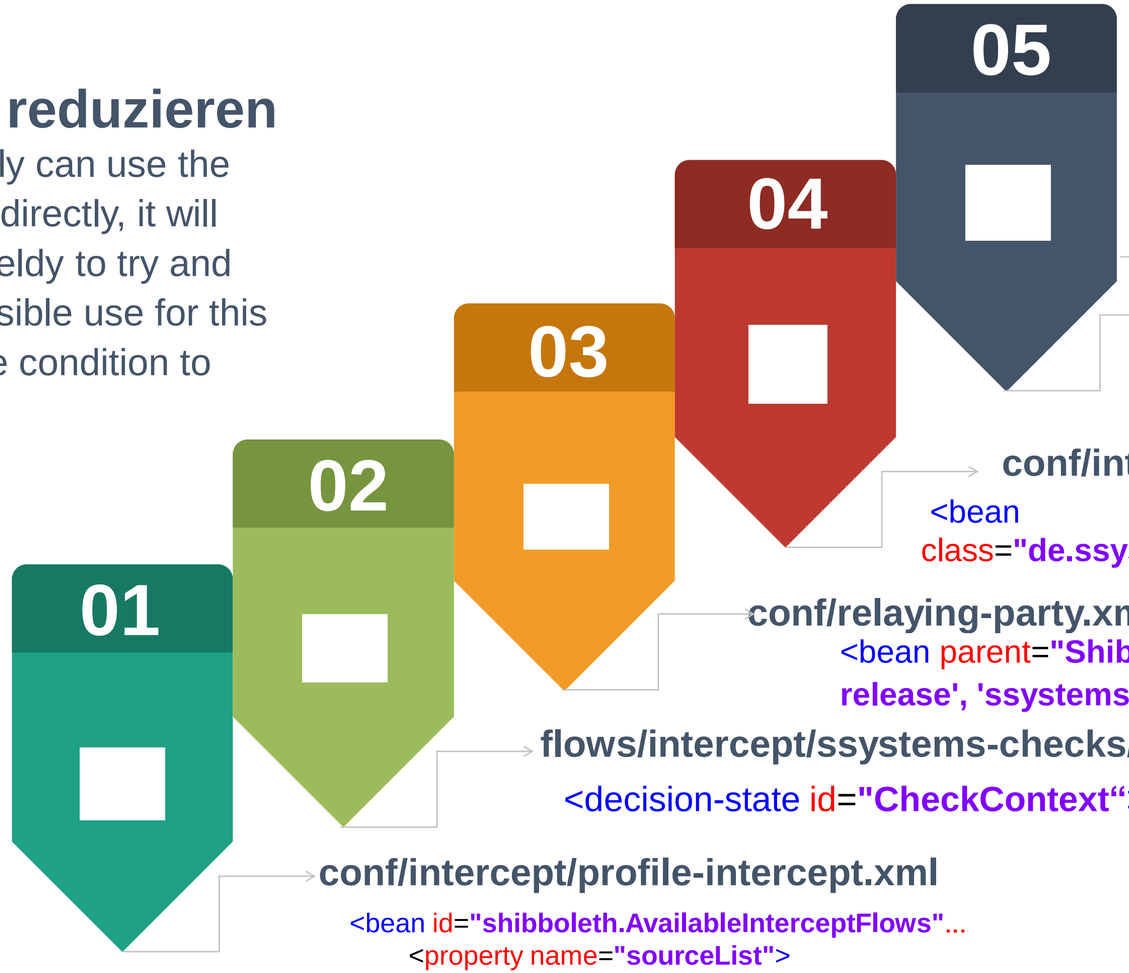
```
/**
 * Predicate that evaluates an {@link
net.shibboleth.idp.attribute.context.AttributeContext} and checks
 * for existance of any attribute. Thus, a filter configuration determines the
behavior.
 *
 */
public class AnyAttributePredicate extends AbstractAttributePredicate
implements InitializingBean {


    @Override
    protected boolean hasMatch(final Map<String, IdPAttribute> attributeMap) {
        log.debug("Checking if attribute map is empty: " + attributeMap);
        return !attributeMap.isEmpty();
    }
```

Eigene Flows

… und eigenes Praedikat

Komplexitaet reduzieren

While you absolutely can use the context-check flow directly, it will often become unwieldy to try and combine every possible use for this feature into a single condition to evaluate.



… /ssystems-checks/ssystems-checks-beans.xml

```
<alias name="shibboleth.ssystems-checks.Condition"
alias="ContextCheckPredicate"/>
```

de/ssystems/…/AnyAttributePredicate.java

```
<decision-state id="CheckContext“>…
        then="proceed" else="ContextCheckDenied" />
```

conf/intercept/ssystems-checks-intercept-config.xml

```
<bean
class="de.ssystems.shibboleth.idp.profile.logic.AnyAttributePredicate"
```

conf/relaying-party.xml

```
<bean parent="Shibboleth.SSO" p:postAuthenticationFlows="#{{'attribute-
release', 'ssystems-checks'}}"/>
```

flows/intercept/ssystems-checks/ssystems-checks-flow.xml

```
<decision-state id="CheckContext“> …. then="proceed" else="ContextCheckDenied" />
```

conf/intercept/profile-intercept.xml

```
<bean id="shibboleth.AvailableInterceptFlows"...
    <property name="sourceList">
      <list merge="true">
+           <bean id="intercept/ssystems-checks“ parent="shibboleth.InterceptFlow" />
```

Scripted Predicate

Sehr flexibel

Konfiguration

- customObject setzen
- Properties auswerten
- Durch boolsche Praedikate beliebig verknuepfbar

```
<bean id="shibboleth.IP.ActivationCondition"

class="net.shibboleth.idp.profile.logic.ScriptedPredicate">
        <property name="customObject"
ref="shibboleth.HttpServletRequest" />
        <constructor-arg>
            <value>
                    // Default return value.
                    var activate = false;

                    // Check the client's IP address.
                    if
(custom.remoteAddr.startsWith("192.168.42.")) {
                        activate = true;
                    }

                    // Return the result as a Boolean object.
                    new java.lang.Boolean(activate);
```

Scripted Predicate

Variabler Context

Contexte finden

- Nach Beans suchen
- Java-Methoden auswerten

```
grep -r bean * | grep shibboleth\. | grep xml:

system/conf/global-system.xml:    <bean id="shibboleth.HttpServletRequest“
system/conf/global-system.xml:    <bean id="shibboleth.HttpServletResponse“
system/conf/global-system.xml:    <bean id="shibboleth.CookieManager“
system/conf/session-manager-system.xml:    <bean id="shibboleth.SessionManager“
system/conf/services-system.xml:    <bean id="shibboleth.AttributeResolverService"
```

Links

Über 70 selbst entwickelte TYPO3-Extensions im Produktivbetrieb

Weitere Informationen

• Context Check Interceptor:
https://wiki.shibboleth.net/confluence/display/IDP30/ContextCheckInterceptConfiguration

• Scripted Praedikat:
https://forge.switch.ch/projects/idp_v3_kerberos_authentication_flow/wiki/SPNEGOAuthnConfiguration/15

•Unit Tests

ssystems

Kastanienalle 32

10435 Berlin

Kontaktieren Sie uns

Ihr Partner für zuverlässige IT-Lösungen

info@ssystems.de

`030202360711

www.ssystems.de