# ssystems

Umfassende, Professionelle IT-Serviceleistungen

## Firmengeschichte

**Gegründet im Jahr 2003 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

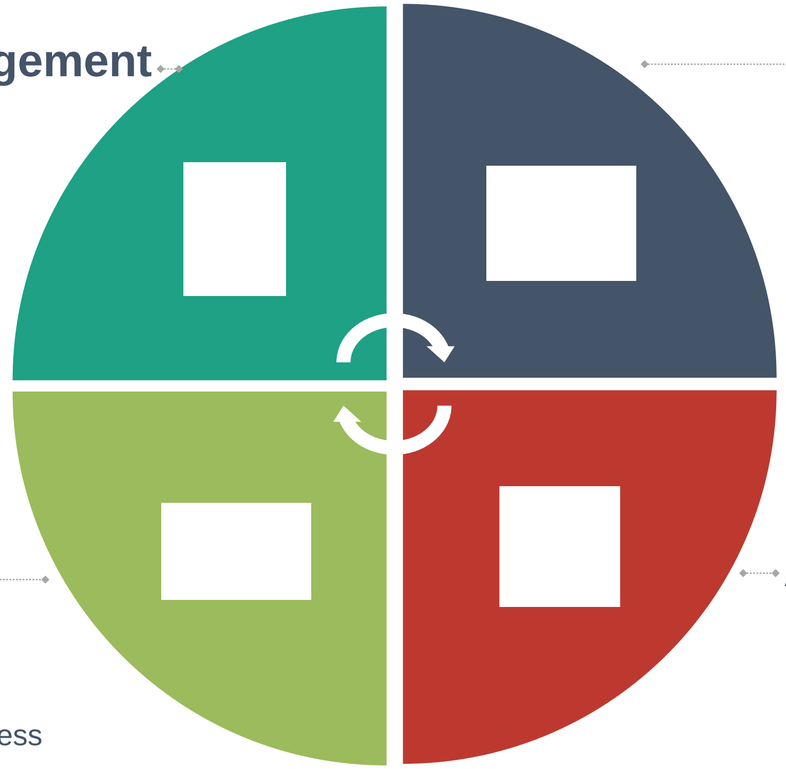
CAMPUSSPEICHER
Hosting mit Geist.

## 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, OpenCAST
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

# Referenzen

Nachhaltige Dienstleistungen

virtuelle hochschule bayern

Bauhaus-Universität Weimar

IUBH International University of Applied Sciences
Internationale Hochschule
School of Business and Management

TECHNISCHE UNIVERSITÄT DRESDEN

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

SIMOVATIVE Campus Management Software

Katholische Stiftungsfachhochschule München
University of Applied Sciences

FRÖBEL Kompetenz für Kinder

HNU HOCHSCHULE NEU-ULM UNIVERSITY OF APPLIED SCIENCES

JG|U UNIVERSITÄTSmedizin. MAINZ

CP Career Partner

HOCHSCHULE ANSBACH

Hochschule Kempten University of Applied Sciences

hochschule aschaffenburg university of applied sciences

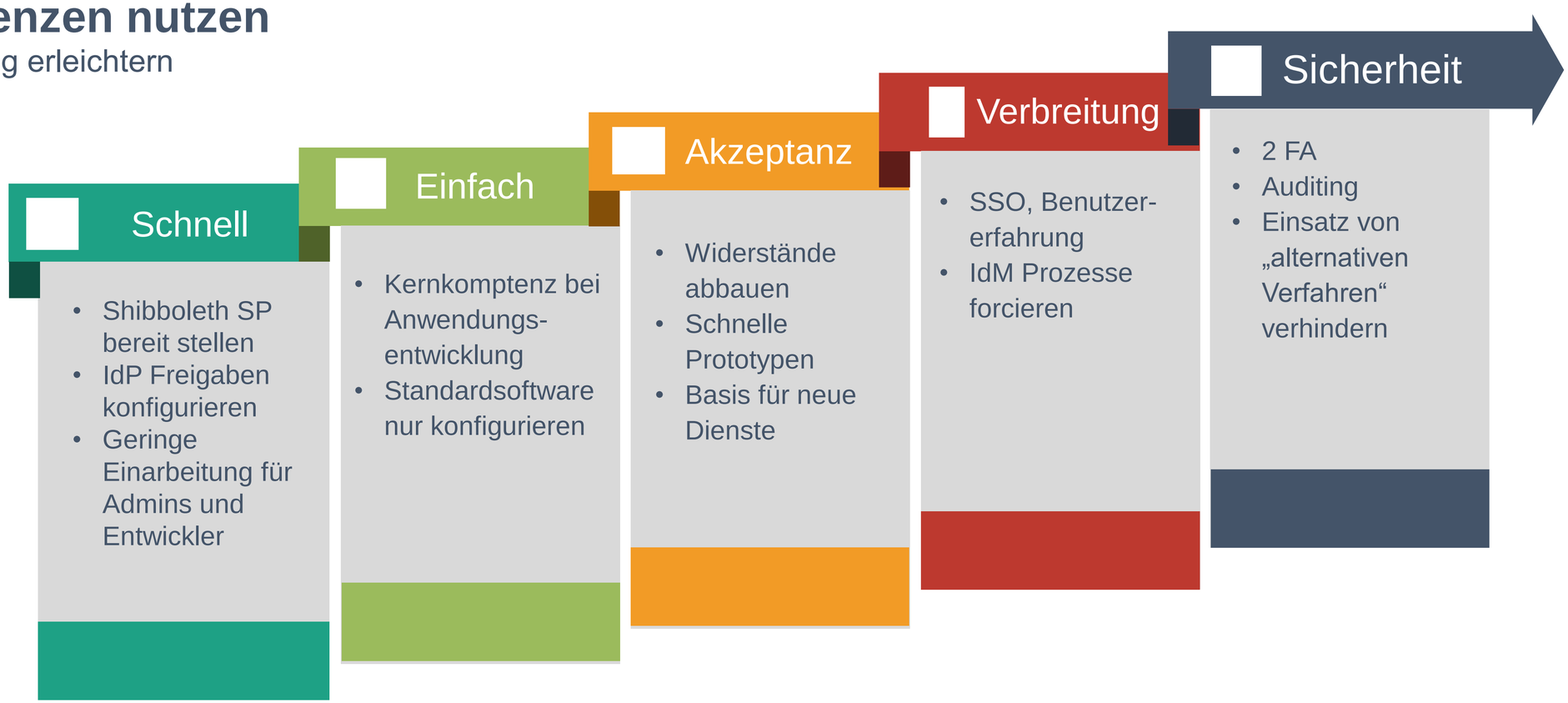
Evangelische Hochschule Nürnberg

# Herausforderung

Automatisierte Verbreitung von AAI

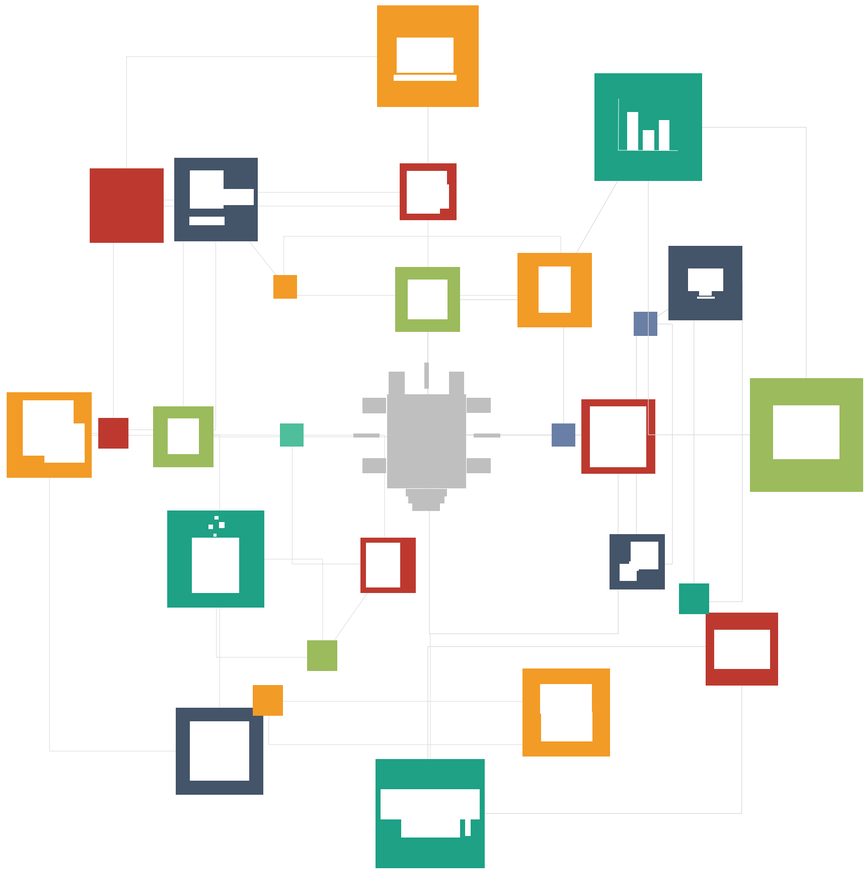
## Komptenzen nutzen

… Einstieg erleichtern



# SP Skills

https://wiki.shibboleth.net/confluence/display/CONCEPT/NativeSPSkills



25%

## OS

standard OS-level knowledge (e.g., how to deal with software installation/uninstallation, manipulate/edit configuration files, start/stop processes, etc.)

25%

## PKI

a basic working knowledge of public key cryptography, similar to the kinds of skills needed to deploy SSL certificates

25%

## XML

a good working knowledge of XML, how to edit and sanity-check it, diagnose parser error messages, etc.

25%

## WWW

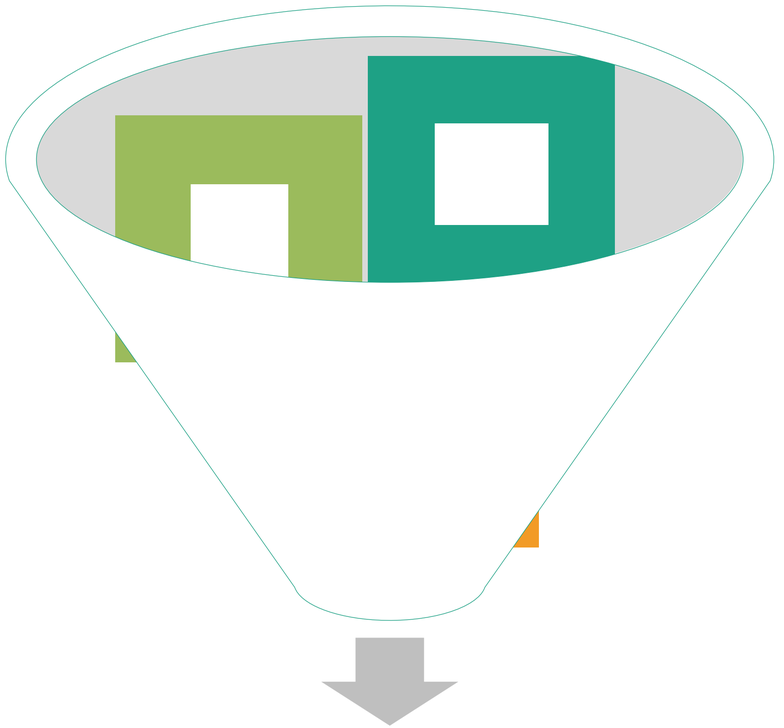
a basic understanding of the web at the level of cookies, redirects, forms, etc. AND a basic understanding of web authentication

# Ein typischer SP

Ziemlich viel Applikationsfremde Technik

## Betriebssystem

- Monitoring
- Log Analyse / Auditing / Alerts
- Backup
- Firewall,…



## Shibboleth SP

- Packages / Repositories
- NTP
- Zertifikate, Apache SSL
- Konfiguration, Attribute Policy und Map

## AAI

- Metadaten registrieren
- Attribute freigeben
- Testuser / Login test

## Application

- Registrierung / Provisionierung
- Authentifizierung
- Deprovsionierung
- Logout

# Warum Saltstack?

Automatisierung der Konfiguration von Serversystemen

**Automatisierung** des Delivery von Shibboleth Service Providern:

Salt is a new approach to infrastructure management built on a dynamic communication bus. Salt can be used for data-driven orchestration, remote execution for any infrastructure, configuration management for any app stack, and much more.

**Pull und Push Verfahren:**
Master/Minion im RZ
Alternativ salt-ssh

CAMPUSSPEICHER
Hosting mit Geist.

**Starke Community**
Shibboleth Formel verfügbar

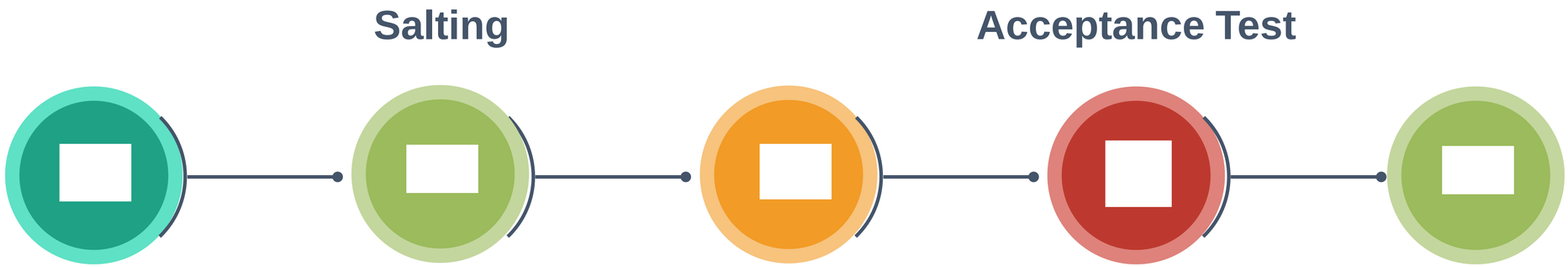
**YAML und Templates.** Einmal verstanden, flexibel einsetzbar.

**Introspection sehr gut:**
Man sieht und versteht gut was passiert

**salt-ssh –i state.apply sp-test.ssystems.de test=True**

# Deployment

Kein Unterschied mehr zwischen Live- und Test-Platformen oder Cluster Nodes




**salt-ssh –i state.apply ‚sp-*.ssystems.de‘ test=False**

# Vorteile für das RZ

Prozesse und Technik verständlich, erlernbar und wiederholbar

## Dokumentation

- Nachvollziehbarkeit von eigenen Tätigkeiten und der von Kollegen
- Konfigurationen zu jedem Zeitpunkt verstehen (GIT) und anwenden

## Delegierbarkeit

- Kollegen müssen in der Lage sein, SPs bereit-zustellen
- Langjährige Shibboleth Erfahrung nicht zwingend notwendig

## Wartung

- Mit vor-konfigurierten VMs nicht möglich
- Zertifikatswechsel
- Ciphers anpassen
- Kleinigkeiten

# Motivaion

ssh key management

## Begriffe:

- State: beschreibt was gemacht wird
- Pillar: beschreibt wie es gemacht wird (Parameter)
- Formel: Menge von States und Pillar Presets

## States

```
ssh_keys_provision_hstrack:
  ssh_auth.present:
    - user: root
    - source: salt://ssh_keys/hstrack.pub
    - config: '%h/.ssh/authorized_keys'


 ssh_keys_deprovision_mitarbeiter:
   ssh_auth.absent:
     - user: root
     - source: salt://ssh_keys/mitarbeiter.pub
     - config: '%h/.ssh/authorized_keys'
```

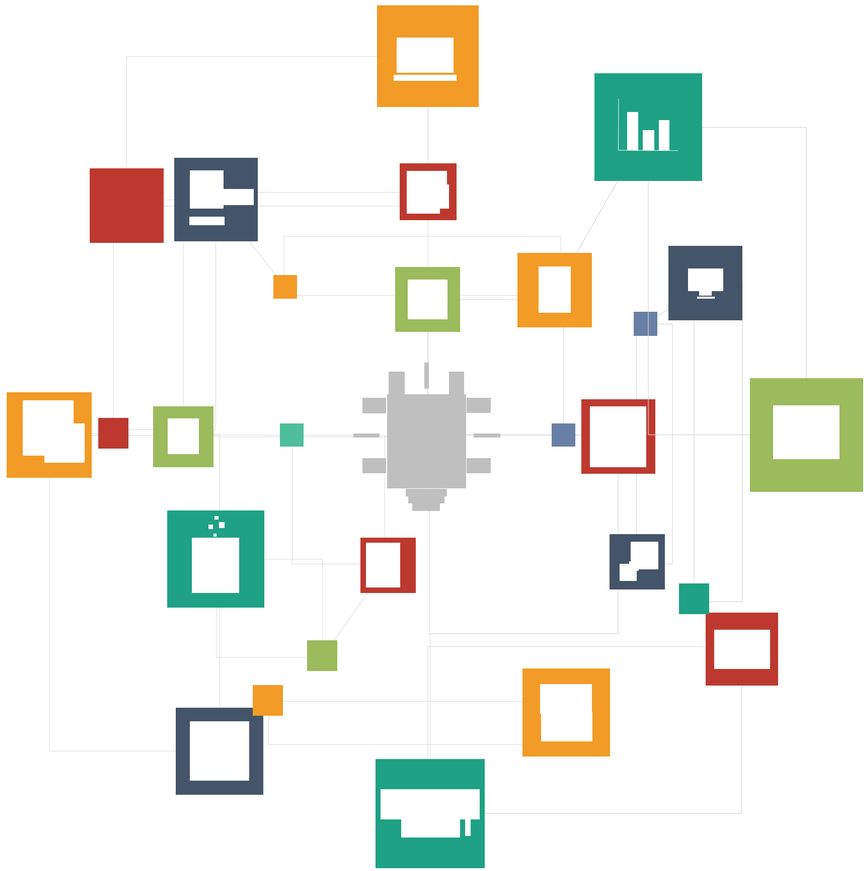
## Mapping von States auf Hosts

```
base:
  '*':
    - ssh_keys/ssh_key_hstrack
    - ssh_keys/ssh_key_deprovision_mitarbeiter #diese state entfernt keys!
    - etckeeper

  '*google.de':
    - ssh_keys/ssh_key_dtrump
    - nginx/ofac
```

# Saltstack SP Formel

https://wiki.shibboleth.net/confluence/display/IDP30/Contributions+and+Extensions



25%

## Entity ID

The SP's SAML entity ID, e.g., https://www.example.com/shibboleth.

25%

## Zertifikate

This key-pair (a DER-encoded X.509 certificate and private key in base64 format) supports inbound XML encryption and signing.

25%

## Sessioninitator

Either unconditionally redirect users to a single identity provider, or send users to a discovery service, where they can choose one of the trusted identity providers.

25%

## Metadata Providers

At a minimum this is a list of URLs, from which the SP can download the XML metadata of trusted identity providers.

# Pillar: Basiskonfiguration

https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example

## Begriffe:

- State: beschreibt was gemacht wird
- Pillar: beschreibt wie es gemacht wird (Parameter)
- Formel: Menge von States und Pillar Presets

```
shibboleth:
  sp:
    entity_id: https://www.example.com/shibboleth
    support_contact: 'webmaster@example.com'
    remote_user:                                                    -
      - eppn
      - upn
      - persistent-id
      - targeted-id
    {%- if grains['osfamily'] != 'RedHat' %}
    backchannel_cipher_suites:  # intermediate-strength profile
      'ECDHE-ECDSA-usw…'
    {%- endif %}
    session_check_address: True
    session_handler_ssl: True
    session_cookie_properties: https
    session_idp_history: True
    sso_default_idp: https://login.example.com/idp/shibboleth
    # sso_discovery_url: https://www.example.com/shibboleth-eds/
    async_logout: False        # workaround for AD FS
```

# Session Handler

https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example

## Begriffe:

- pillar.example: Gesamtkonfiguration, in jeder Formel enthalten

```
session_handlers:
    - type: MetadataGenerator
      location: /Metadata.xml
      signing: False        # TODO: configure signing                    -
    - type: Status
      location: /Status
      acl:
        - 192.0.2.0/24      # example management network
        - 127.0.0.1         # IPv4 loopback (default)
        - ::1               # IPv6 loopback (default)
    - type: Session
      location: /Session
      show_attribute_values: True
    - type: DiscoveryFeed
      location: /DiscoFeed
```

# Metadaten

https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example

## Metakonfiguration:

- Lokale Metadaten
- DFN Metadaten

```
metadata_providers:
    - type: XML
      url: https://sso.example.de/metadata/Local-AAI-metadata.xml
      max_refresh_delay: 3600                                          -
    - type: XML
      url: https://www.aai.dfn.de/fileadmin/metadata/sha2/DFN-AAI-metadata.xml
      max_refresh_delay: 3600
      legacy_org_names: True
      metadata_filters:
          - type: Signature
            certificate: |
                -----BEGIN CERTIFICATE-----
                    <gekuerzt>
                -----END CERTIFICATE-----
```

# Zertifikate

https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example

## SP Zertifikat und Key

- Können inline oder in extra Dateien hinterlegt werden
- Getrennte Zertifikate für Encryption und Signing möglich

```
shibboleth:
  sp:
    encryption_certificate: &spcert |
        -----BEGIN CERTIFICATE-----
        ...
        -----END CERTIFICATE-----
      encryption_key: &spkey |
        -----BEGIN PRIVATE KEY-----
        ...
        -----END PRIVATE KEY-----
      signing_certificate: *spcert
      signing_key: *spkey
```

# States: SP und Apache

https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example

## Eigene States:

- SP und NTP werden vollautomatisch durch Formel konfiguriert
- Apache hier durch eigene States

```
include:
  - shibboleth.repo
  - shibboleth.sp
  - ntp

httpd_config_sp_vhost:
  file.managed:
    - name: /etc/httpd/conf.d/{{ grains[id'] }}.conf
    - template: jinja
    - source: {{ salt['pillar.get']('httpd:ssl-vhost-template','salt://etc/httpd/conf.d/ssl-vhost.conf') }}
    - user: root
    - group: root
```

**salt-ssh –i state.apply sp-test.ssystems.de test=True**

# Zertifikate in der Schleife

## Jinja Templates:

- Variablen
- Schleifen
- If-Then-Else
- WWW-Technologie!

```
{% set defaultDomain   =  salt['grains.get']('id') %}
{% set domain_names = salt['pillar.get']('ssl-certificates:domain_names', [defaultDomain]) %}
{% for domain in domain_names %}
/etc/pki/tls/certs/{{ domain }}-cert-nginx.pem:
  file.managed:
    - source: salt://etc/pki/tls/certs/{{ domain }}-cert-nginx.pem
    - user: root
    - group: root
    - mode: 644

/etc/pki/tls/certs/{{ domain }}-cert.pem:
  file.managed:
    - source: salt://etc/pki/tls/certs/{{ domain }}-cert.pem

/etc/pki/tls/certs/{{ domain }}-chain.pem:
  file.managed:
    - source: salt://etc/pki/tls/certs/{{ domain }}-chain.pem

/etc/pki/tls/private/{{ domain }}-key.pem:
  file.managed:
    - source: salt://etc/pki/tls/private/{{ domain }}-key.pem
{% endfor %}
```

# Maps: HTTPs, Ciphers

## Noch mehr Apache:

- Redirect auf HTTPs, dynamisches vhost Template
- Sichere TLS Konfiguration
- Kompatible Ciphers (v.a. Java clients)

```
{% from "httpd/map.jinja" import httpd with context %}
{{httpd.vhost_conf}}:
  file.managed:
    - template: jinja
    - source: {{ salt['pillar.get']('httpd:settings:vhost-                    -
template','salt://etc/httpd/conf.d/redirect-ssl-vhost.conf') }}
    - user: root
    - group: root

httpd_config_bettercrypto:
  file.managed:
    - name: {{httpd.bettercrypto_conf}}
    - template: jinja
    - source: {{ salt['pillar.get']('httpd:bettercrypto-
template','salt://etc/httpd/conf.d/bettercrypto.ssl') }}
    - user: root
    - group: root
```

# States: Pitfalls und Backup

## Kleinigkeiten:

Turn this on to support "require valid-user" rules from other mod_authn_* modules, and use "require shib-session" for anonymous session-based authorization in mod_shib.

```
{% from "httpd/map.jinja" import httpd with context %}
shibd_compat_valid_user_on:
  file.replace:
    - name: {{httpd.shibd_conf}}
    - pattern: ShibCompatValidUser Off
    - repl: ShibCompatValidUser On

download_netbackup_client:
  archive.extracted:
    - name:   /opt/
    - source: https://syncandhsare.de/NetBackup_7.7.2_CLIENTS2.tar
    - unless: ls /opt/NetBackup_7.7.2_CLIENTS2

install_netbackup:
  cmd.run:
    - cwd: /opt/NetBackup_7.7.2_CLIENTS2/
    - name: /opt/NetBackup_7.7.2_CLIENTS2/install
    - user: root
    - unless:
      - ls /usr/openv/netbackup/bin/bp
```

# Teilkonfiguration

https://docs.saltstack.com/en/latest/ref/states/all/salt.states.file.html

**Wartbare Konfiguration:**

- Auf Basis von Markern warden Inhalte in Dateien gepatcht
- {{vhb_idp_entity_id}} wird durch Test- bzw. Live IdPs ersetzt

```
shibboleth_sp_patch_always_aggregation:
  file.blockreplace:
    - name: {{ shibboleth2_xml }}
    - marker_start: '<!-- MARKER START -->'
    - marker_end: '<!-- MARKER END -->'                                    -
    - content: |
        <AttributeResolver type="SimpleAggregation" attributeId="schacPersonalUniqueCode"
format="urn:oid:1.3.6.1.4.1.25178.1.2.14">
            <Entity>{{vhb_idp_entity_id}}</Entity>
            <EntityReference>External-Links</EntityReference>
            <saml2:Attribute xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
FriendlyName="eduPersonEntitlement" />
        </AttributeResolver>
```

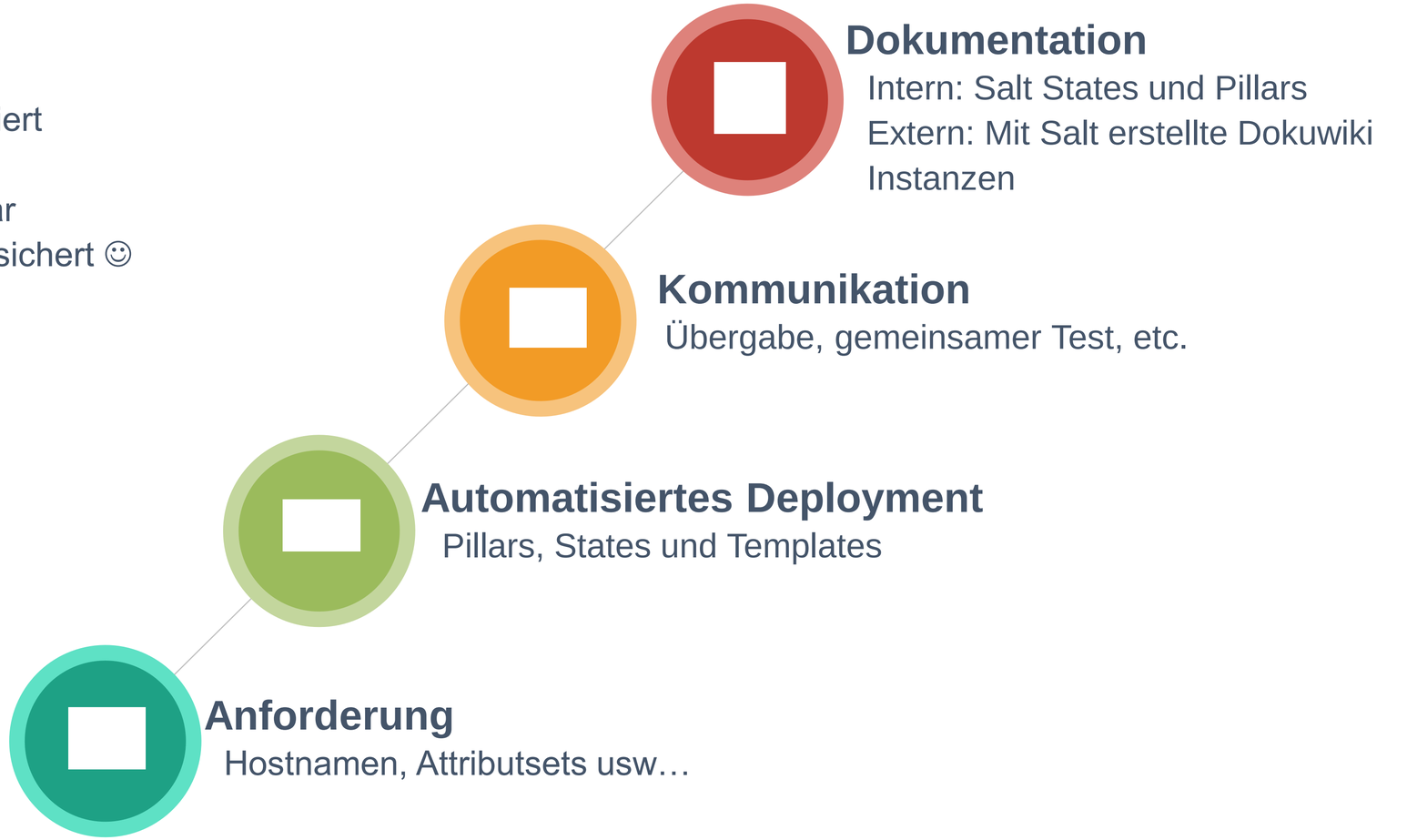
# Zusammenfassung

Abstrakter Vorgang jenseits der Technik

## Einfacher Prozess

**100%**

… dokumentiert
… wartbar
… delegierbar
… Urlaub gesichert ☺



# Ausblick: IdP

https://github.com/irtnog/shibboleth-formula

## IdP Deployment:

- Hoch komplexe Konfigurationen kann man nur mit hoch komplexen Metakonfigurationen beschreiben
- In Teilen nutzbar

```
idp:
  hostname: login.example.com
  entity_id: https://login.example.com/idp/shibboleth
  scope: example.com
  cookie_secure: yes
  keystore_password: 'longalphanumericpassword'
  sealer_password: 'adifferentlongalphanumericpassword'

  ## Deploy Shibboleth IdP using the same service account as the
  ## desired Java servlet container
  user: tomcat
  group: tomcat

  packages:
    - java-1.8.0-openjdk-devel
    - bash
    - tomcat
```

# Ausblick: Tomcat

https://github.com/saltstack-formulas/tomcat-formula

## Tomcat Deployment:

- Formel basiert

```
tomcat:
  # The Tomcat version can be overridden
  ver: 7
  security: 'no'
  java_opts:                                                    -
    - 'Didp.home=/opt/shibboleth-idp‘  …
  with_haveged: true
  haveged_enabled: true
  cluster:
    simple: true

  connectors:
    ajp:
      port: 8009
      address: '127.0.0.1'
      protocol: 'AJP/1.3'
      maxPostSize: 100000
      maxThreads: 500
      …
```

# Ausblick: Tomcat

https://github.com/saltstack-formulas/tomcat-formula

## Tomcat Deployment:

- IdP wird hier in Tomcat vhost betrieben
- Lohnt sich schon für Stage- und Live-Systeme sowie für Cluster

```
sites:
    idp.vhost:
      name: localhost
      appBase: ../webapps/idp
      alias: {{ grains['id'] }}
      host_parameters:
        unpackWARs: "true"
        autoDeploy: "true"
        deployXML: "false"
      path: 'idp'
      docBase: '/opt/shibboleth-idp/war/idp.war'
      unpackWAR: "true"
      antiResourceLocking: "true"
      swallowOutput: "true"
      privileged: "true"
      autoDeploy: "true"
      debug: 0
```

# Ausblick: IdP

## ECP für O365:

- O365 kann nur eine Domain pro IdP Entity ID
- Dynamisch erzeugte ECP endpoints für virtuelle IdPs notwendig

```
{% if salt['pillar.get']('idp:ecp:enabled') == True %}
{% set ecp_domain_names = salt['pillar.get']('idp:ecp:domain_names') %}
{% for domain in ecp_domain_names %}
        Alias /domains/{{ domain }}/ECP /idp/profile/SAML2/SOAP/ECP
        <Location /domains/{{ domain }}/ECP>
          AuthType Basic
          AuthName "Shibboleth Identity Provider - {{ domain }} ECP profile"
          AuthBasicProvider ldap
          AuthLDAPRemoteUserAttribute uid
          AuthLDAPURL "{{ AuthLDAPURL }}?mailRoutingAddress,uid?sub?(mailAccessDomain={{ domain }})"
          AuthLDAPBindDN "{{ AuthLDAPBindDN }}"
          AuthLDAPBindPassword "{{ AuthLDAPBindPassword }}"
          #      Require all granted
          require valid-user
         ProxyPreserveHost On
         RequestHeader set SHIB_ECP_REMOTE_USER %{REMOTE_USER}s
         ProxyPass ajp://localhost:8009/idp/profile/SAML2/SOAP/ECP
         ProxyPassReverse ajp://localhost:8009/idp/profile/SAML2/SOAP/ECP
        </Location>
{% endfor %}
```

# Dokumentation

Beratung und Support

SALTSTACK

https://saltstack.com/

Frei verfügbar!

## Weitere Informationen:

- https://github.com/irtnog/shibboleth-formula/blob/master/pillar.example
- https://docs.saltstack.com/en/latest/topics/best_practices.html