Technische Universität Berlin

TRESOR Tutorial

Mathias Slawik | Service-centric Networking | Rendez-Vouz | Time

TELEKOM INNOVATION LABORATORIES

Service-centric NETWORKING

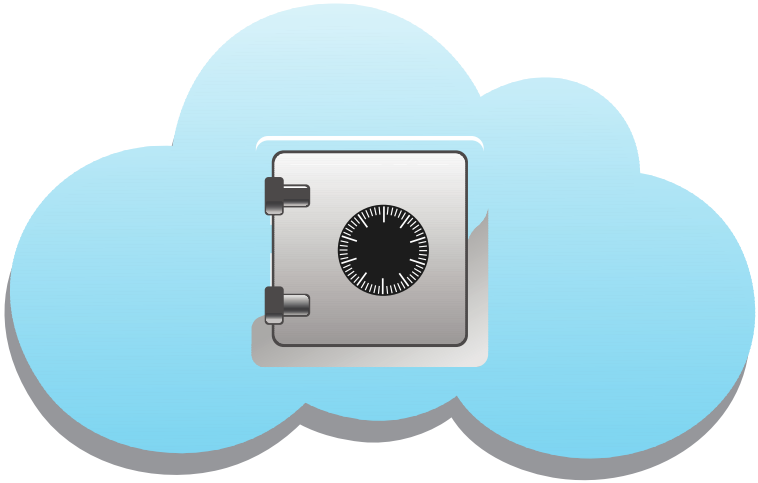
Agenda

- Introduction to TRESOR
- TRESOR components in depth
- Live demo of TRESOR ecosystem
- Outlook and next steps
- Q&A

Technische Universität Berlin

INTRODUCTION TO TRESOR

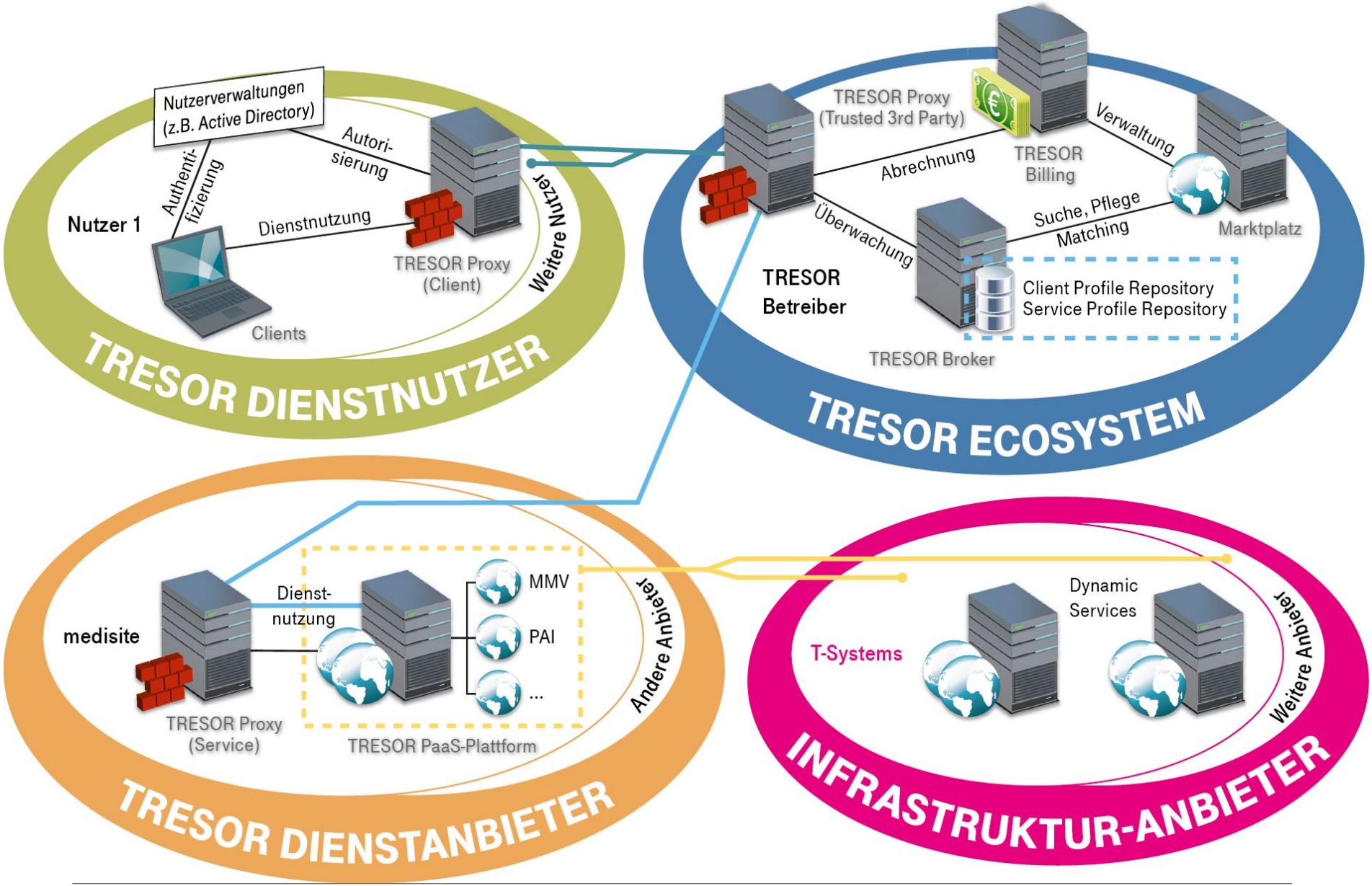
TRESOR - www.cloud-tresor.[de|com]



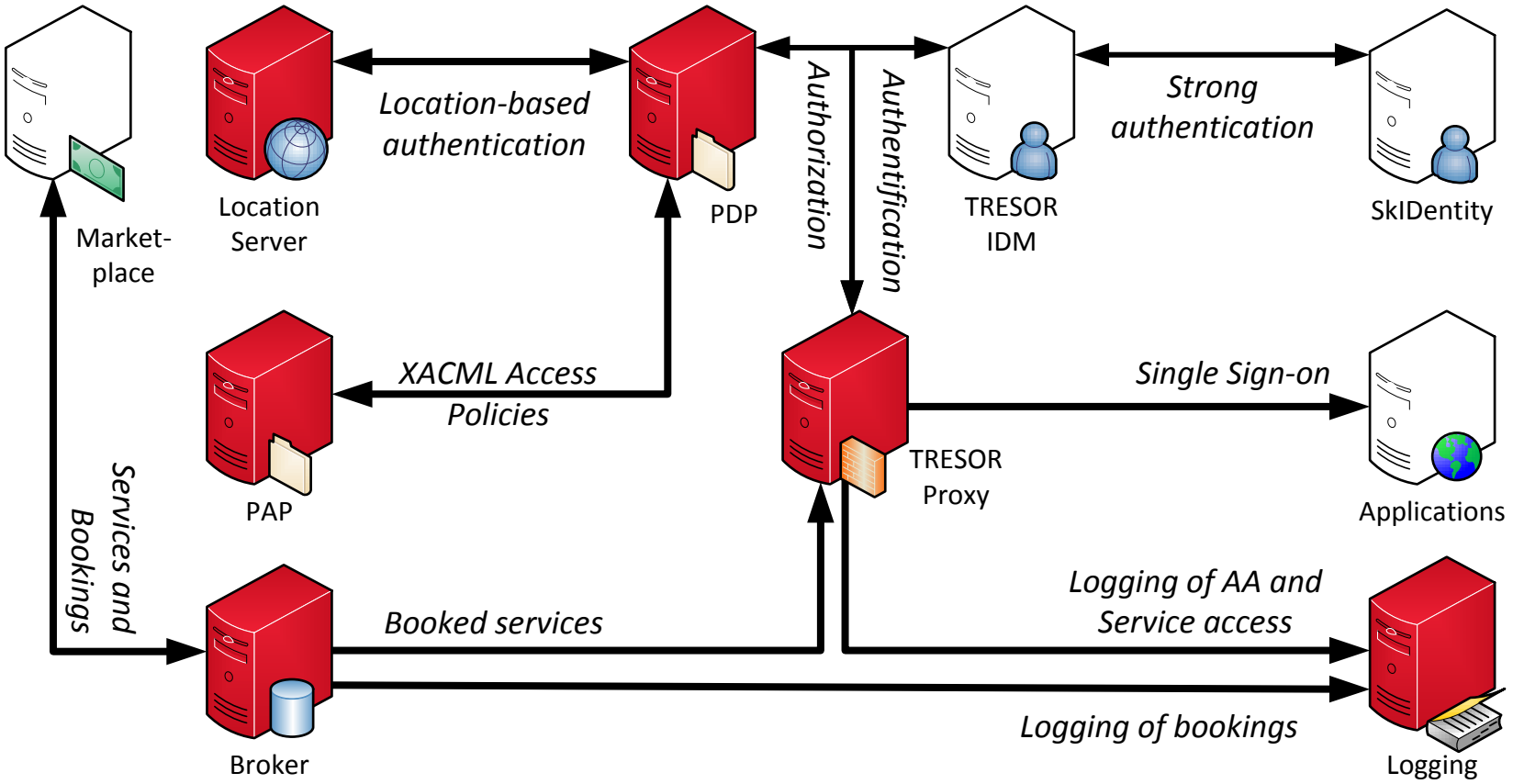
- **Tr**usted **E**cosystem for **S**tandardized and **O**pen cloud-based **R**esources
  - Funded by BMWi, runs from 03.2012 until 03.2015
- Building a Secure Cloud Ecosystem for the German Healthcare sector

TRESOR - www.cloud-tresor.[de|com]

- Technical / Development Partners
  - medisite (project lead)
  - T-Systems International & T-Systems Multimedia Solutions
  - TUB Service-centric Networking (SNET)
- Application Partners
  - German Heart Institute / Paulinenkrankenhaus
  - TUB Information and Communication Management (IKM)
  - Bitplaces


Nutzerverwaltungen
(z.B. Active Directory)
Authenti-
fizierung
Autori-
sierung
Nutzer 1
Dienstnutzung
TRESOR Proxy
(Client)
Clients
Weitere Nutzer
TRESOR DIENSTNUTZER
TRESOR Proxy
(Trusted 3rd Party)
TRESOR
Billing
Abrechnung
Verwaltung
Überwachung
Suche, Pflege
Matching
Marktplatz
TRESOR
Betreiber
Client Profile Repository
Service Profile Repository
TRESOR Broker
TRESOR ECOSYSTEM
medisite
Dienst-
nutzung
MMV
PAI
...
TRESOR Proxy
(Service)
TRESOR PaaS-Plattform
Andere Anbieter
TRESOR DIENSTANBIETER
T-Systems
Dynamic
Services
Weitere Anbieter
INFRASTRUKTUR-ANBIETER

SNET Components



This repository Search Explore Gist Blog Help mathiasslawik

TU-Berlin-SNET / tresor-ecosystem

Unwatch 7 Star 0 Fork 0

Run the TRESOR ecosystem as docker containers — Edit

12 commits 1 branch 0 releases 1 contributor

branch: master tresor-ecosystem / +

added logging stack

Ömer Malik Ilhan authored 13 days ago latest commit 7dbab54b88

components	added logging stack	13 days ago
config	added logging stack	13 days ago
data	Added TRESOR PAP	20 days ago
.gitignore	Added TRESOR PAP	20 days ago
.gitmodules	updated submodule	13 days ago
LICENSE	Initial commit	25 days ago
README.md	Prepare production deployment	25 days ago
fig.yml	added logging stack	13 days ago

Code
Issues 0
Pull Requests 0
Wiki
Pulse
Graphs
Settings

HTTPS clone URL
https://github.com/

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop
Download ZIP

README.md

tresor-ecosystem

Run the TRESOR ecosystem as docker containers

Open Source Repository

https://github.com/TU-Berlin-SNET

- TRESOR Components
  - TRESOR Ecosystem, TRESOR Broker, SDL-NG, TRESOR Proxy, TCTP Implementation, Demo-Apps, TRESOR PDP
- Soon
  - PAP, Location Server & Client

Technische Universität Berlin

TRESOR COMPONENTS IN DEPTH

Components

- TCTP
- Proxy
- Authentication
  - Claims-based federated identity
- Authorization
  - XACML Policy Decision Point (PDP)
  - XACML Policy Administration Point (PAP)
- Broker & SDL-NG

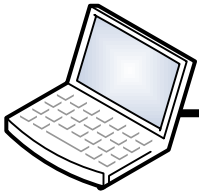
The Trusted Cloud Transfer Protocol

TCTP

Trusted Cloud Transfer Protocol (TCTP) Management Summary

- Novel HTTP security mechanism
- Complements HTTPS
- Additional Security for SaaS
- Advances the state-of-the-art considerably

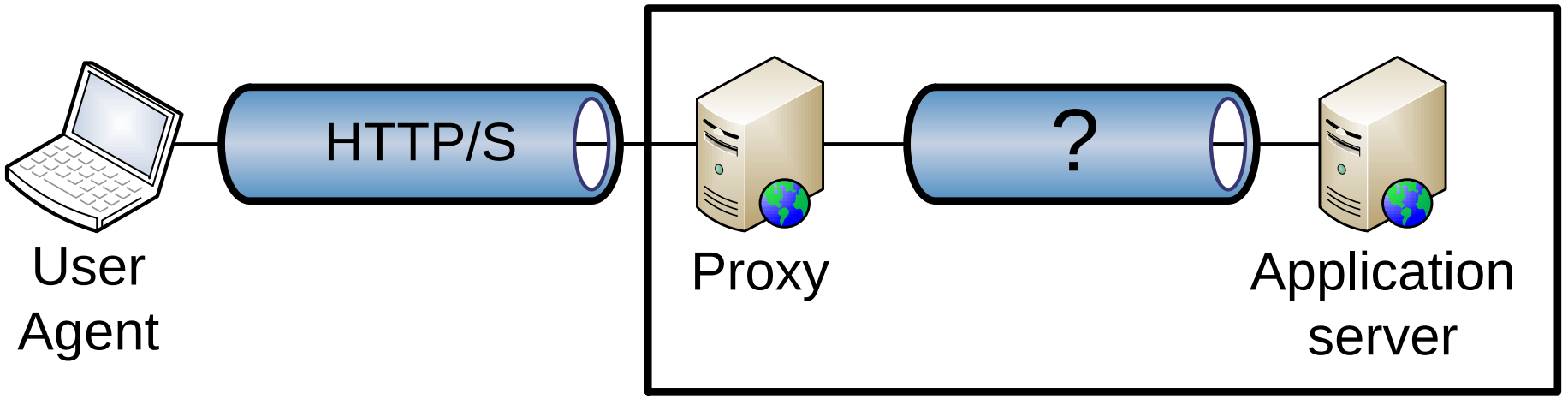
HTTP communication (in TRESOR practice)



User Agent

SaaS App

Challenge: Implementing HTTP intermediaries



a) Relay TLS?

- ☑ Confidential plaintext
- ☒ No HTTP management

b) Serve TLS?

- ☑ HTTP management
- ☒ Available plaintext

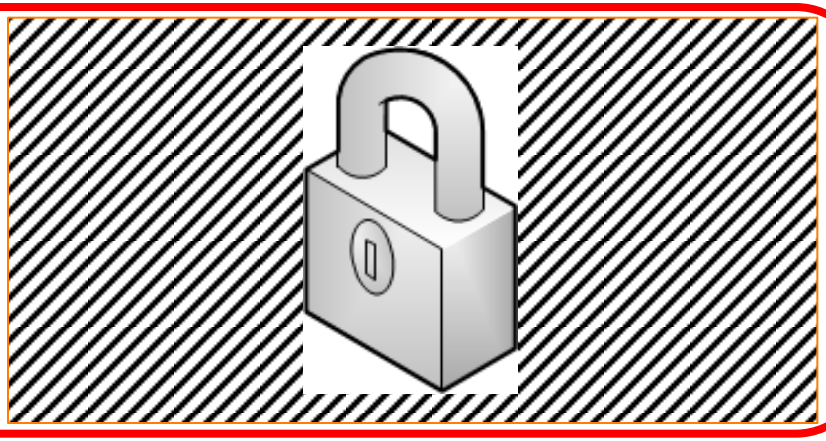
Solution = HTTP entity-body encryption

Technische Universität Berlin

Less confidential
Needed for HTTP mgmt.

Often confidential
Not needed for HTTP mgmt.

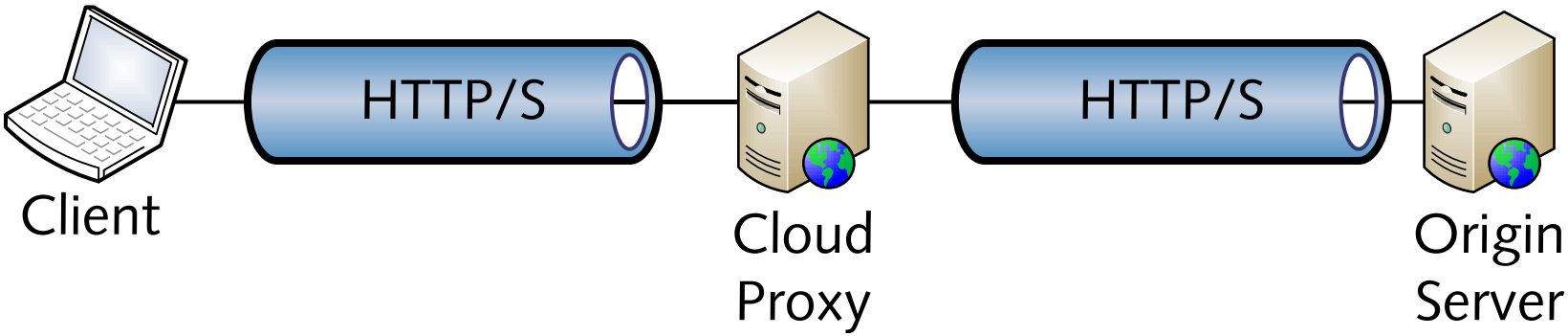
```
POST /patients HTTP/1.1↩
Content-Type: text/json↩
Content-Length: 81↩
↩
```



TCTP: General Idea

- TCTP = Exchanging TLS over HTTP
- **H**TTP **A**pplication **L**ayer **E**ncryption **C**hannels
  - Persist TLS session state
  - Required for multiple connections
  - HTTP semantics for TLS Handshake & Encryption
- General Process
  1. End-to-end key exchange (TLS Handshake Protocol)
  2. HTTP entity body encryption (TLS Records)

Key exchange



POST /halecs HTTP/1.1↩ ↩					
CT	V	L	CT	V	L
Handshake			Handshake		

TCTP: How does it look?

Technische Universität Berlin

Unencrypted header fields allow HTTP management	POST /patients HTTP/1.1↩ Content-Type: text/json↩ Content-Length: 81↩ Content-Encoding: encrypted↩ ↩
HALEC URL	/halecs/1Mfjk941xkFe↩
Encrypted TLS Records contain HTTP body	ñ½┘ ╓änÖiz«╥ñ¦wÆÉ,± »_)S╩(@oⁿ┘└└╙ ┌╖ nµGÄ_ └╩Q %"£┬N¼ƒ╢╬∩•&i

Capability discovery

```
OPTIONS * HTTP/1.1↩
Accept: text/prs.tctp-discovery↩
↩
```

```
HTTP/1.1 200 OK↩
Content-Type: text/prs.tctp-discovery↩
Content-Length: 81↩
↩
/:↩
/(service(.+?))?:↩
/(service(.+?)/)?static.*:↩
/(service(.+?)/)?.*:/\1/halecs
```

First client request

POST on discovered HALEC creation URL.

```
POST /halecs HTTP/1.1↩
Content-Length: 211↩
↩
```

TLS Record client_hello

```
 ╬  ╩Rîç[±Ülû
Kfóu╣ºΩ:Ɣ±t├ëx d8π╩}U  \LL
9 8 ê çLL 5 äLL  LL LL  3
2 Ü Ö E DLL / û A LLLL
                D

 4 2
#
```

Server response

URL of new HALEC

TLS Records:
ServerHello, Certificate,
ServerKeyExchange,
ServerHelloDone

```
HTTP/1.1 200 OK↩
Content-Length: 1050↩
Location: /halecs/Adaw7VXdVpu↩
↩
 5  1RîÇ[ymP‰9Ñ_z-
└cêN╝>╔▒_P‰J E4pr╧  9
#  ¥Ö
ô0éÅ0éâ  000131120095643Z131
120105643Z0,10Utctp-
server10Æ&ëô≥,dtctp0é"0*åHå≈
é 0éé╖┬
"!çë‖}Ç æÉA€)ßd▒≤╡≤Ü)■n...
```

Second client request

POST on newly created HALEC URL.

```
POST /halecs/Adaw7VXdVpu HTTP/1.1↩
Content-Length: 198↩
↩
```

TLS Records: ClientKeyExchange, ChangeCipherSpec, Finished

```
 å  é ÇΣZë½E╒)U ╪3╘6ßò¥
,▌4å╩<e>Ö_•■≤{╣5¿AµP¼/3çæyαD╘╓├
Z!q}÷gàhV*┴M│Yo ∞¦.wê═╫3°<7MJ•╤
!ó.=µ▄òm3┬g=)IHƒ╥íiΩ\▒âï╢T•
06Fnq#πºeb≡┌
Hàv¢├v¢FΣw┤±ÑmFÖ?°?[iqiî_⌂╪`∞ar
JQ
```

Server response

TLS Records:
ChangeCipherSpec,
Finished

```
HTTP/1.1 200 OK↩
Content-Length: 266↩
↩
 ╨   ╞        └┴G•ù«δA╨▦╕  ¥°ø▒û
qA≤0N&ô╖R¿tX"ΣWαP‰Id┌
√/C]╨?╫└ǫ╞ⁿ#┼¬δ{  *Y╨┤G≥Dô
e.╨╤{+!=`M÷─Ö╫ò{²┌Γαë  òh1ç└
Wq7g╕α        L•╨j┌L╠Ex┤╠δçû
RdBª┼╔ôé*ºJez\`&ⁿvµ═╕σ=▒6╨V
╪%tY}P▐╨÷Fô╬"É┐~╕O≈╖αïV',⌐Ö
└▓U╨0┌┤\╨e╠Æ  ╙•$Æσ╨╠&;d╕J┤µ
╔╢•0/╫/YUE";ⁿ°ù9ì┴≤αtJ
```

Enhancements wrt related work (S/MIME, XML Encryption & Signature, ...)

- Efficient binary message format
- Additional HTTP „message-flow“ security
- Streaming capability
- Efficient implementation
- HTTP compatible

Technische Universität Berlin

Secure Cloud Proxy

PROXY

The TRESOR Proxy

- Modular HTTP proxy
  - Written in Ruby (using extensions in C/C++)
  - Open Source (Apache 2.0 license)
- Functionality
  - TCTP Client & Server
  - Reverse/Forward proxying
  - Claims-based Authentication
  - XACML-based Authorization
  - Routing for booked services
  - Logging & Monitoring

Client Proxy	Central Proxy	Service Proxy	Any Proxy
HTTP Reverse	HTTP Reverse	HTTP Reverse	HTTP Reverse
HTTP Forward	HTTP Forward	HTTP Forward	HTTP Forward
TCTP Client	TCTP Client	TCTP Client	TCTP Client
TCTP Server	TCTP Server	TCTP Server	TCTP Server
Single-Sign-On	Single-Sign-On	Single-Sign-On	Single-Sign-On
TLS Cert-Auth	TLS Cert-Auth	TLS Cert-Auth	TLS Cert-Auth
XACML	XACML	XACML	XACML
Mon/Log/Aud	Mon/Log/Aud	Mon/Log/Aud	Mon/Log/Aud
REST-API/MQ	REST-API/MQ	REST-API/MQ	REST-API/MQ
			Plugin A
			Extension B

github.com/TU-Berlin-SNET/tresor-proxy

Running the proxy

Technische Universität Berlin

Via Docker

```
docker run -i -t mathiasslawik/tresor-proxy <proxy options>
```

Manually, as a ruby script

```
bundle exec ruby bin/proxy.rb <proxy options>
```

Manually, as a daemon

```
bundle exec ruby bin/proxy_daemon.rb <command> ↩
<daemon options> -- <proxy options>
```

Proxy configuration

```
Usage: proxy.rb [options]
-b, --broker            The URL of the TRESOR broker
-i, --ip                The ip address to bind to (default: all)
-p, --port              The port number (default: 80)
-n, --hostname          The HTTP hostname of the proxy (default: proxy.local)
-P, --threadpool        The Eventmachine thread pool size (default: 20)
-t, --trace             Enable tracing
-l, --loglevel          Specify log level (FATAL, ERROR, WARN, INFO, DEBUG - default INFO)
    --logfile           Specify log file
-C, --tctp_client       Enable TCTP client
-S, --tctp_server       Enable TCTP server
    --tls               Enable TLS
    --tls_key           Path to TLS key
    --tls_crt           Path to TLS server certificate
    --reverse           Load reverse proxy settings from YAML file
    --raw_output        Output RAW data on console
    --sso               Perform claims based authentication
    --xacml             Perform XACML
    --pdpurl            The PDP URL
    --fpurl             The SSO federation provider URL
    --hrurl             The SSO home realm URL
-h, --help              Display this help message.
```

Functionality (1)

- TCTP Client & Server
  - TCTP discovery mechanism
  - TCTP key exchange
  - TCTP encryption
- Forward proxying
  - Regular “proxy” functionality
- Reverse proxying, configured through
  - reverse.yml file (Host -> Upstream host)
  - TRESOR Broker integration (endpoint URLs of booked services)

Functionality (2)

- Claims-based Authentication
  - Redirection of unauthenticated users
  - Relaying authenticated user information as HTTP headers
  - Using SAML-tokens for authorization of requests
- XACML-based Authorization
  - Based on URL, HTTP action, user attributes, user organization, accessed service

Functionality (3)

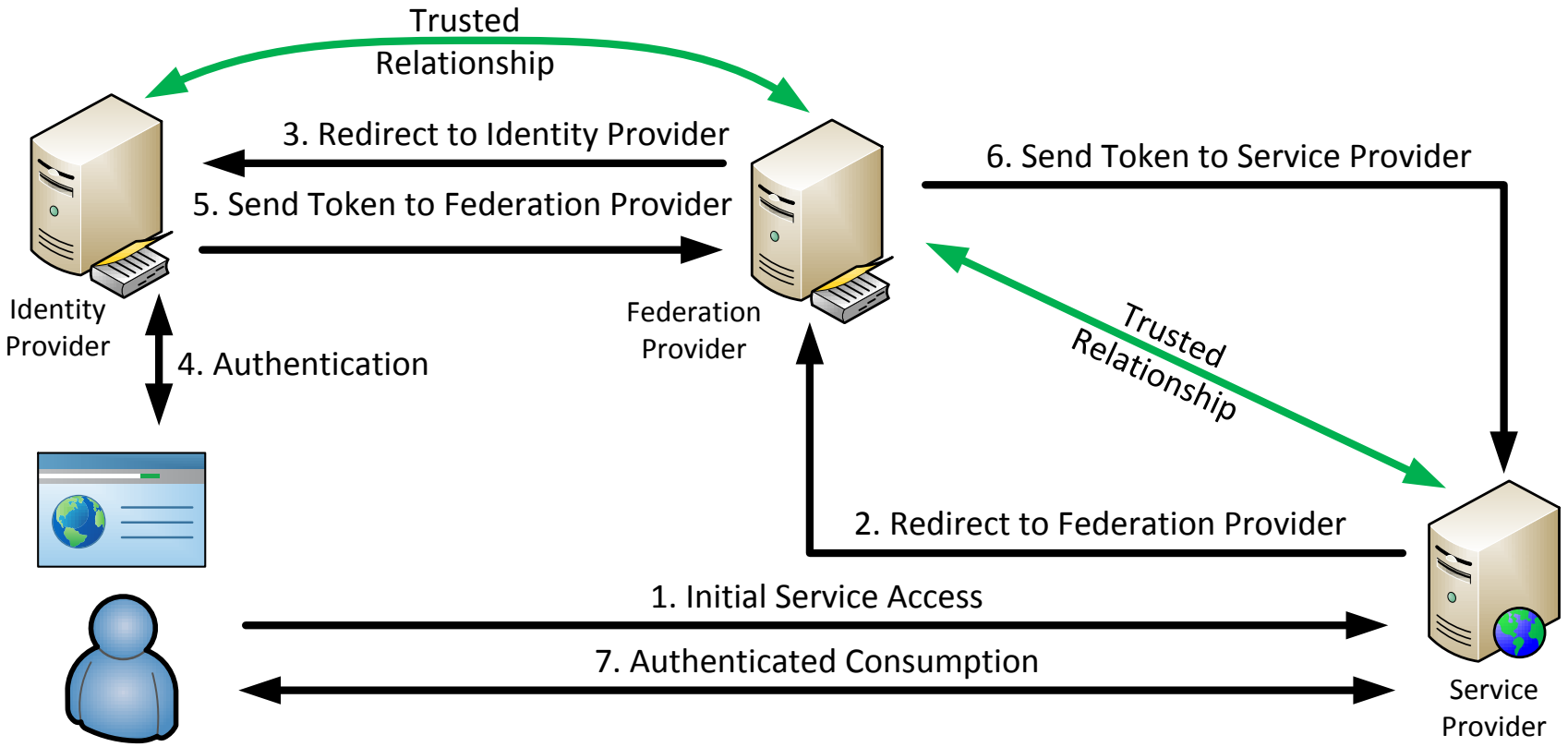
- Routing for booked services (in TRESOR broker)
  - Requests are routed to the specific endpoint of the specific deployment of a specific user organization
- Logging & Monitoring

Technische Universität Berlin

Claims-based Identity

AUTHENTICATION

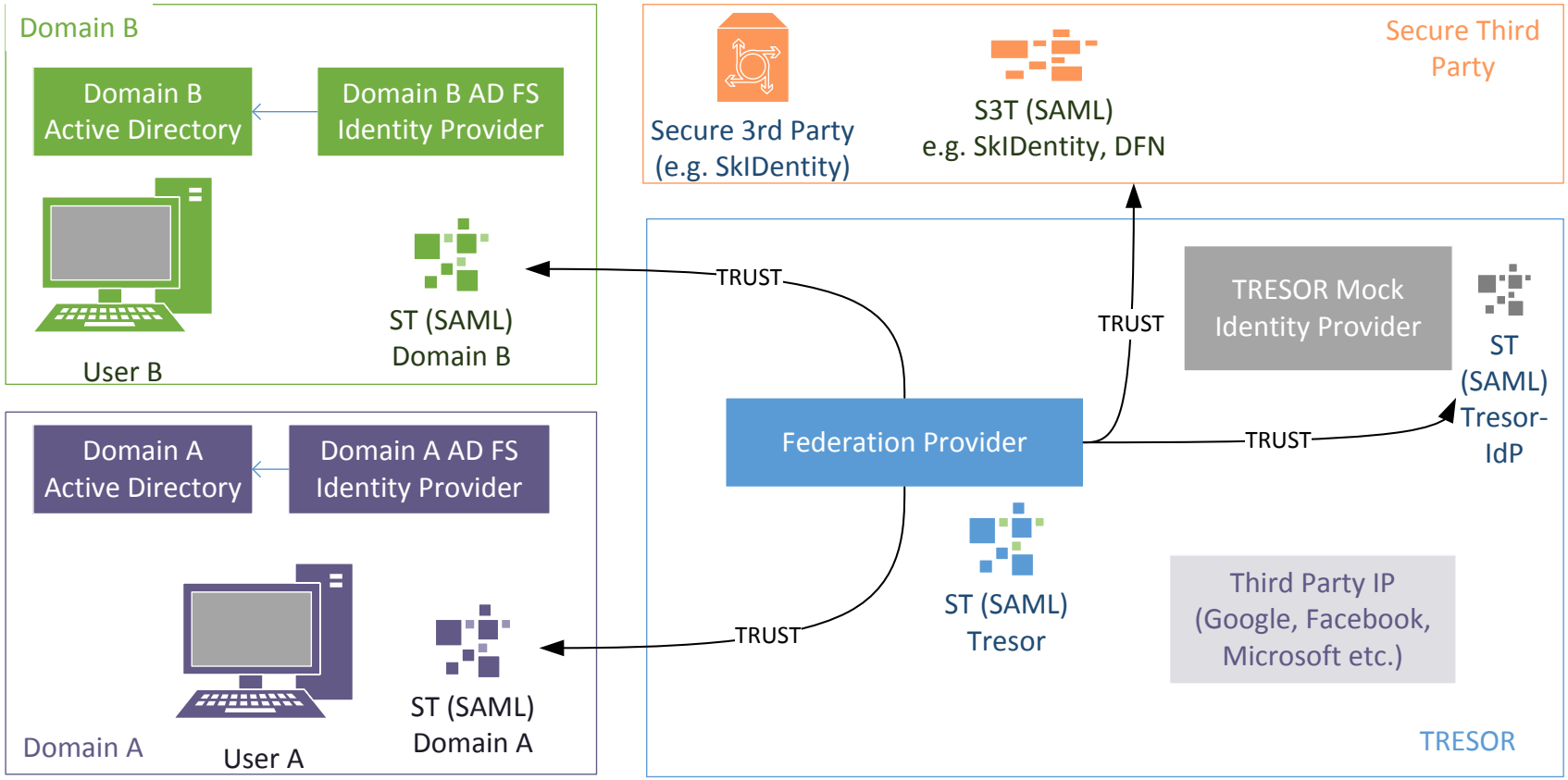
Federated Authentication Flow



TRESOR Testbed (publicly accessible)

- Microsoft Active Directory Federation Services
- Identity Providers
  - ASP.NET Mock Identity Provider
  - SkIDentity (Secure SmartCard Authentication)
  - Demo Active Directories
- Planned
  - Replacement by Open Source Federation Services and Identity Providers (e.g. RedHat PicketLink)

TRESOR Testbed architecture



SAML – Token example

spec/support/test_sso_token.erb

(in Proxy source)

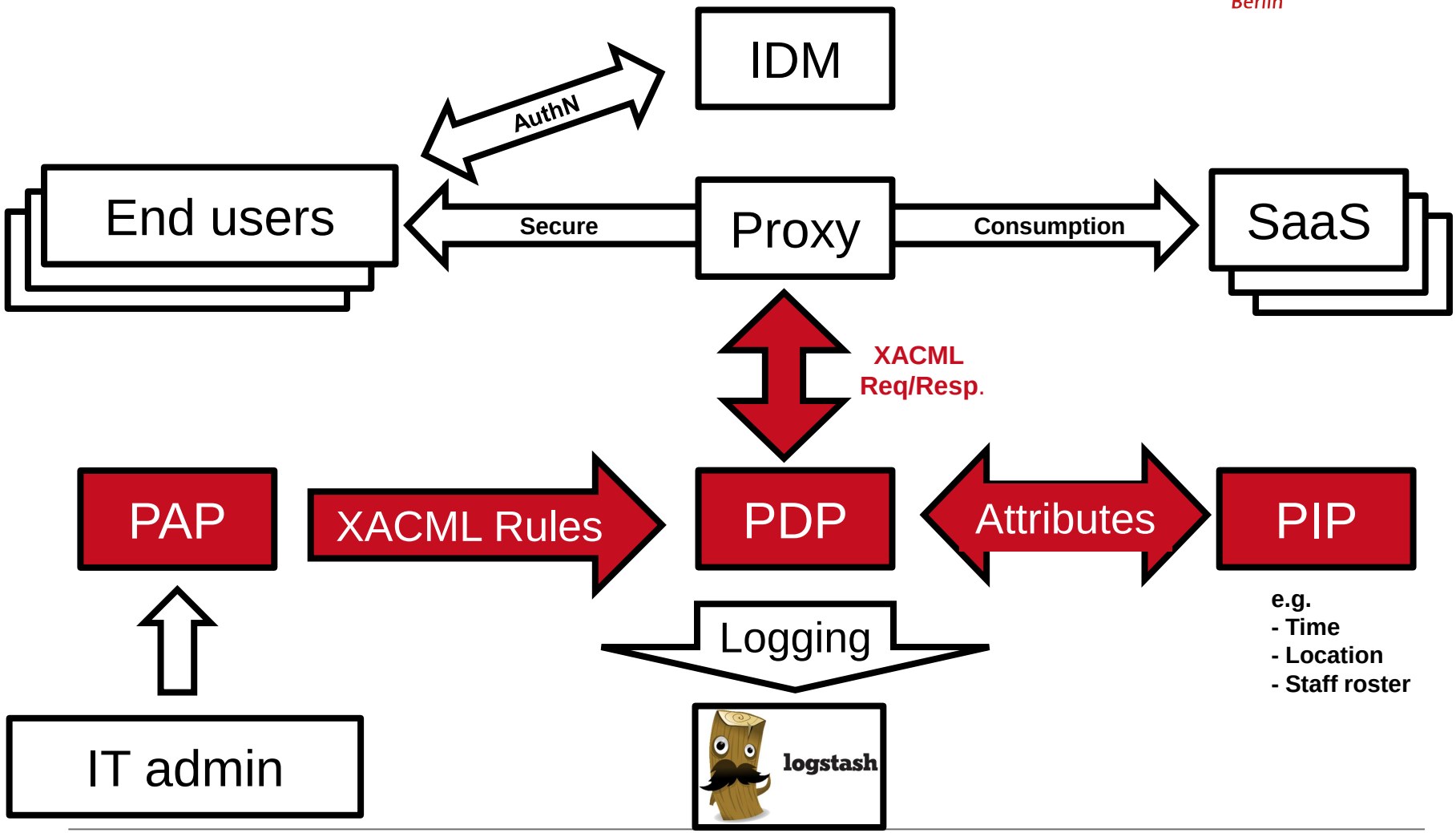
XACML Policy Decision Point (PDP) and
XACML Policy Administration Point (PAP)

AUTHORIZATION

Goals

- Definition of authorization rules by cloud *users* instead of application-specific rules
  - Basis: RESTful URLs
- Integration of external information, e.g.
  - Location, staff roster, schedule of responsibilities, etc.
- Compliance to legal provisions

Distributed Authorization



Policy Decision Point

- Basis: WSO2 Balana
- Additions
  - Multi-tenant (e.g., federated) and multi-service policy store
  - „Attribute finders“ for user location, doctor/nurse roster, ...
- RESTful interface
  - CRUD Rules
  - Retrieving XACML decisions

XACMLv3 Request

- Access Subject
- Resource
- Action

XACML Request
Category: Access Subject

- Subject ID
  - taken from the SAML token
- Organization UUID
  - queried from the TRESOR broker using the organization name from the SAML token
- Subject Attributes
  - e.g. user groups, taken from the SAML token

XACML Request
Category: Resource

- Service UUID
  - queried from the TRESOR broker using the first part of the hostname, e.g., "tresordemo" from "tresordemo.service.cloud-tresor.com"
- Resource ID
  - the request path, e.g., "/patients/8172391" or "/admin"

XACML Request
Category: Action

- Action ID
  - the HTTP method, e.g. "GET" or "DELETE"

Examples

- XACML Requests
  - lib/tresor/frontend/xacml/xacml_request.erb in proxy source
- XACML Rules
  - TRESOR ecosystem example (later)

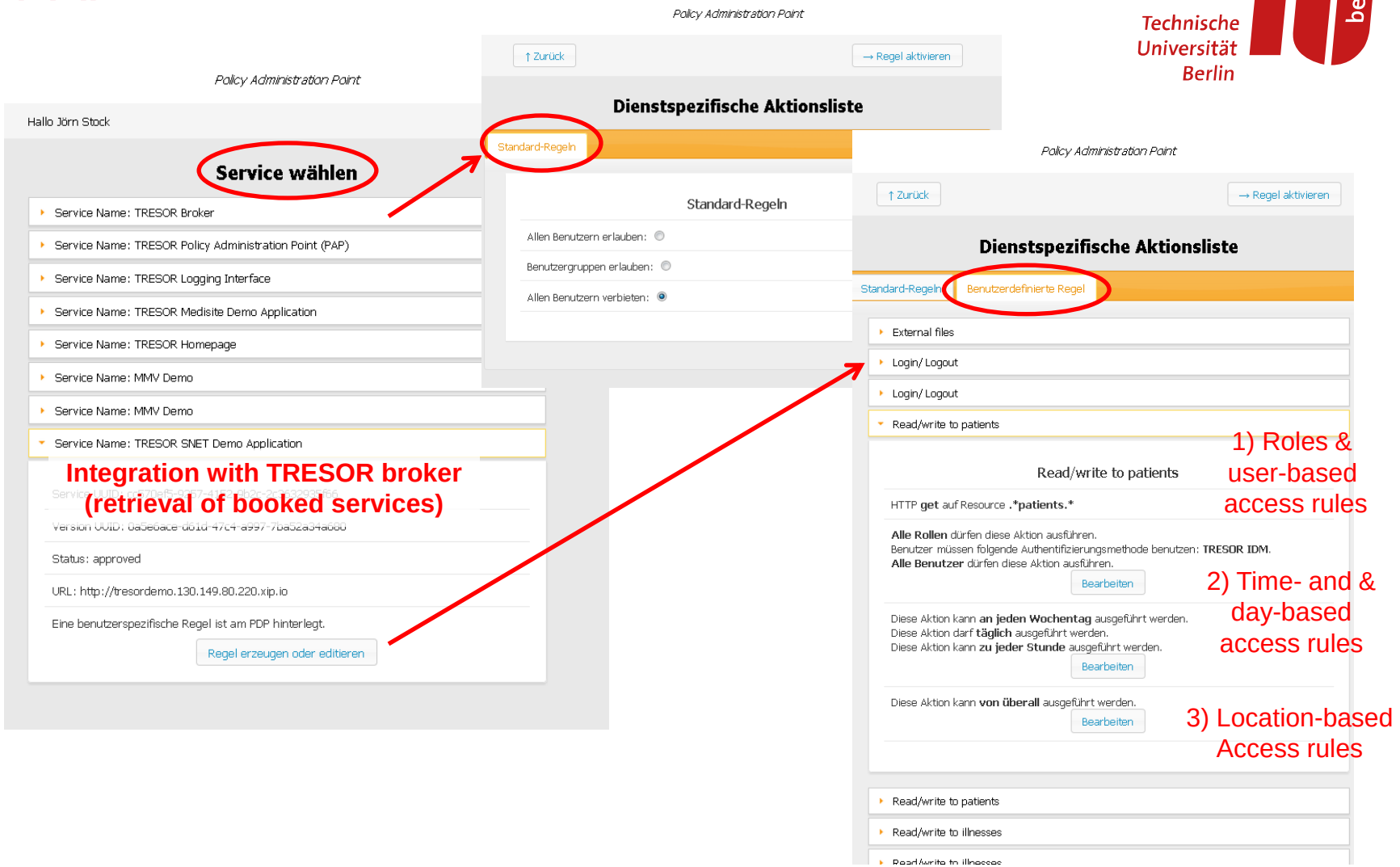
Policy Administration Point (PAP)

- Interface for creating RESTful XACML rules
- Development by Philip Raschke (TRESOR student)
- Published:
  Raschke, P. and Zickau, S. (2014). A Template-Based Policy Generation Interface for RESTful Web Services. *On the Move to Meaningful Internet Systems: OTM 2014 Workshops*. Springer Berlin Heidelberg, 137-153.

Policy Administration Point: Main Idea

1. Applications provide RESTful routes as XML
   - Route description, URL path regexp, HTTP methods & parameters
2. Admins combine routes with access conditions
   - User name, user group, location, time
3. Created rules are uploaded to the PDP
   - Uses Service/Organization combination for multi-tenancy

PAP


Policy Administration Point
Hallo Jörn Stock
Service wählen
Service Name: TRESOR Broker
Service Name: TRESOR Policy Administration Point (PAP)
Service Name: TRESOR Logging Interface
Service Name: TRESOR Medisite Demo Application
Service Name: TRESOR Homepage
Service Name: MMV Demo
Service Name: MMV Demo
Service Name: TRESOR SNET Demo Application
Integration with TRESOR broker
(retrieval of booked services)
Status: approved
URL: http://tresordemo.130.149.80.220.xip.io
Eine benutzerspezifische Regel ist am PDP hinterlegt.
Regel erzeugen oder editieren
Policy Administration Point
↑ Zurück
→ Regel aktivieren
Dienstspezifische Aktionsliste
Standard-Regeln
Standard-Regeln
Allen Benutzern erlauben:
Benutzergruppen erlauben:
Allen Benutzern verbieten:
Policy Administration Point
↑ Zurück
→ Regel aktivieren
Dienstspezifische Aktionsliste
Standard-Regeln
Benutzerdefinierte Regel
External files
Login/ Logout
Login/ Logout
Read/write to patients
Read/write to patients
HTTP get auf Resource .*patients.*
Alle Rollen dürfen diese Aktion ausführen.
Benutzer müssen folgende Authentifizierungsmethode benutzen: TRESOR IDM.
Alle Benutzer dürfen diese Aktion ausführen.
Bearbeiten
Diese Aktion kann an jeden Wochentag ausgeführt werden.
Diese Aktion darf täglich ausgeführt werden.
Diese Aktion kann zu jeder Stunde ausgeführt werden.
Bearbeiten
Diese Aktion kann von überall ausgeführt werden.
Bearbeiten
Read/write to patients
Read/write to illnesses
1) Roles & user-based access rules
2) Time- and & day-based access rules
3) Location-based Access rules

Technische Universität Berlin

Describing Services

BROKER & SDL-NG

Management Summary

- Service marketplaces lack automation
- Academic approaches lack business relevance, simplicity and maturity
- TRESOR approach enables automation, while being simple and pertinent for businesses

TRESOR Broker - User stories

- As an SME cloud provider, I want to describe my SaaS offerings
  - My “Health Record” service saves its data in Germany, can import Word and PDF files, and can be used by Internet Explorer and Chrome
- As an SME cloud consumer, I want to search for services matching my selection criteria
  - Show me all services, which save their data in the EU, can import PDFs and can be used by Chrome
- As an SME marketplace operator, I want to define a vocabulary for service description
  - Let marketplace stakeholders describe their data location, importable formats, compatible browsers, etc.

TRESOR Broker – SME Constraints

- *Support* vs. *automation*
- Limited expertise
- Central: economic viability
- „Good enough“ instead of highest sophistication
- Current Marketplace: two prototypical services

TRESOR Broker
Functional requirements

1. **Be pertinent to businesses**
   - Vocabulary (selection criteria), employed technology, user interfaces, ...
2. **Prevent redundant information**
   - Allow for reuse of documentation in existing websites, etc.
3. **Support consumer service selection**
   - Extensive documentation, rationale, guidelines, ...

TRESOR Broker
Non-functional requirements

1. **Low description effort**
   - Intuitive language, high comprehensibility
2. **Low language definition effort**
   - Allow easy adaptation of vocabulary for different scenarios
3. **Simple and usable tooling**
   - Simple (meta-)model, simple editor, test-covered software

Components

1. SDL-NG framework
2. Pertinent business vocabulary
3. TRESOR broker

Open Source: https://github.com/TU-Berlin-SNET

Challenge area 1

A textual domain-specific language

for **tooling simplicity**

Domain-specific languages

- “a computer programming language of limited expressiveness focused on a particular domain” [11]
- External DSL
  - regular expressions, SQL, XML
- Internal DSL
  - fluid APIs, C++ templates

SDL-NG framework examples

```
type :cloud_service_model
cloud_service_model :saas
cloud_service_model :paas
cloud_service_model :iaas
cloud_service_model :haas

type :charge_unit
charge_unit :user_account
charge_unit :floating_license


service_properties do
  string :service_name

  cloud_service_model

  charge_unit :is_charged_by

  add_on_repository do
    url
    integer :number_of_add_ons
  end
end
```

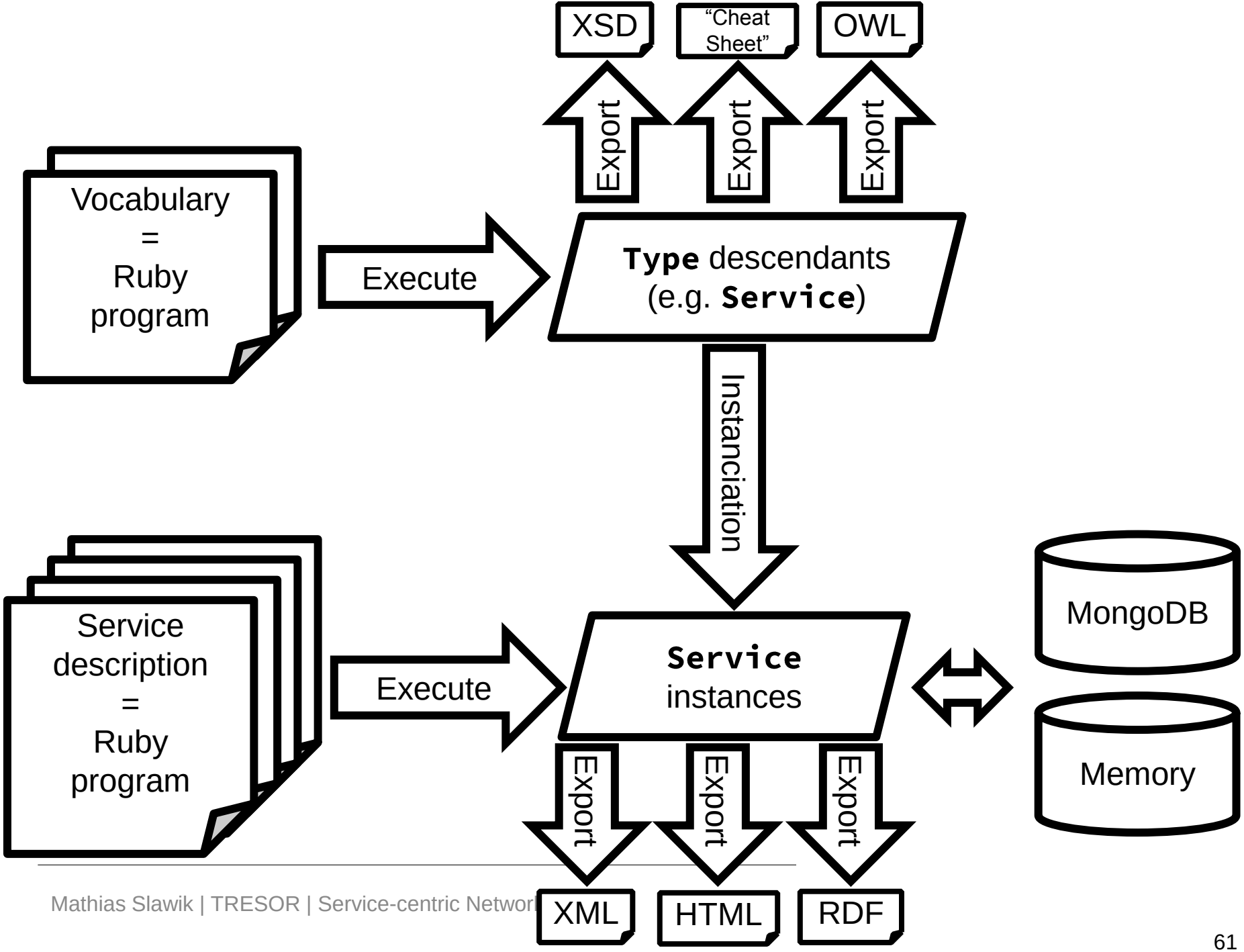
Vocabulary definition

```
service_name 'Google Drive for Business'

cloud_service_model saas
add_on_repository do
  url 'https://www.google.com/enterprise/marke
  number_of_add_ons 1000
end

is_charged_by user_account
```

Service Description


XSD
"Cheat Sheet"
OWL
Export
Export
Export
Vocabulary = Ruby program
Execute
Type descendants (e.g. Service)
Instanciation
Service description = Ruby program
Execute
Service instances
MongoDB
Memory
Export
Export
Export
XML
HTML
RDF

Challenge area 2

A dynamic DSL

for **reusing documentation**

SDL-NG framework – dynamic DSLs

- Language = Ruby program code
  - Dynamically adaptable
- Easy inclusion of existing sources
  - Amazon Spot-market prices
  - Cloud exchanges (e.g. DBCE)
  - HTML descriptions
- Blend static information & dynamic content

SDL-NG dynamic DSL example

Technische Universität Berlin

```
dynamic do
  fetched_features = fetch_from_url 'http://www.salesforce.com/sales-cloud/overview',
 '.slide h3 + *'

  feature 'Mobile', fetched_features[0]
  feature 'Contact Management', fetched_features[1]
  feature 'Opportunity Management', fetched_features[2]
  feature 'Chatter', fetched_features[3]
  feature 'Email Integration', fetched_features[4]
end
```

Features	Name	Mobile
	Description	Now, Salesforce, combined with your custom apps and AppExchange mobile apps, turns your mobile device into a portable sales office. You can log calls, respond to hot leads, work opportunities, or check dashboards no matter where you are. And collaborate across teams, from anywhere.
	Name	Contact Management
	Description	Bring social intelligence to your sales process. Gain insights from popular social media sites — like Facebook, Twitter, LinkedIn, YouTube, and Klout — right within Salesforce to help you increase productivity.
	Name	Opportunity Management

Challenge area 3

A vocabulary, based on empiric research,
for **business pertinence**

- Current sources:
  - “Cloud Requirement Framework” by Repschläger, et al.
  - “Cloud Service Check” by Value4Cloud
  - “Service Measurement Index” by CSMI

Pertinent business vocabulary

Area	Example properties
Characteristics	Cloud service model, service categories
Charging	Charge unit (user account, floating license)
Compliance	Data location, audit options (e.g. audit logging)
Delivery	Billing and payment options
Dynamics	Duration for provisioning an end user
Interop	Features, interfaces, and compatible browsers
Optimizing	Maintenance windows and future roadmaps
Portability	Exportable and importable data formats
Protection	Communication protection (HTTPS, VPN)
Provider management	Support availability
Reliability	Offline capabilities
Reputation	Year of service establishment
Trust	Providers' financial statement, reference customers

= 37 `Type` classes, 31 `Service` properties and 52 `Type` instances

TRESOR Broker

- Integrates all components
- Ruby on Rails, MongoDB

TRESOR Broker
Authoring

- JavaScript-based editor for service descriptions
- „Cheat sheet“ for easy reference
- RESTful CRUD actions, if authored externally

TRESOR Broker
Service editor

TRESOR Broker
Cheat sheet

Technische Universität Berlin

Service

A service

Properties

`service_name`	Service name (SDLString)
`cloud_service_model`	Cloud service model (CloudServiceModel)
`service_categories[]`	Categories (ServiceCategory)
`service_tags[]`	Tags (SDLString)
`add_on_repository`	Service Add-On Repository (AddOnRepository)
`is_charged_by`	Charging unit (ChargeUnit)
`data_location`	Data location (Location)
`data_deletion_policy`	Data deletion policy (SDLUrl)
`status_page`	The status page URL (SDLUrl)

TRESOR Broker
Cheat sheet

Technische Universität Berlin

CloudServiceModel

A cloud service model

Predefined instances

`:saas`	Software as a Service (SaaS)
`:paas`	Platform as a Service (PaaS)
`:iaas`	Infrastructure as a Service (IaaS)
`:haas`	Hardware as a Service (HaaS)

CommunicationProtection

A communication protection means

Predefined instances

`:https`	HTTPS encryption
`:vpn`	VPN connection

TRESOR Broker
Cheat sheet

Technische Universität Berlin

Provider

A service provider

Properties

`provider_name`	Name (SDLString)
`provider_location`	Location (SpecificLocation)
`company_type`	Company type (CompanyType)
`employs`	Number of employees (SDLNumber)
`partner_network`	Partner network (SDLUrl)
`reference_customers[]`	Reference customers (ReferenceCustomer)
`last_years_revenue`	Last years revenue (SDLMoney)
`initiatives[]`	Participatory initiatives (Initiative)
`reports[]`	Published reports (ReportingInformation)

TRESOR Broker - Interfaces

- RESTful interface
  - CRUD Services, Service versions, Clients, Providers
  - Booking of services
  - Retrieving services in different formats (SDL-NG, XML/XSD, HTML, RDF)
- Browser interface currently in development
  - Part of „Open Service Compendium“ / my dissertation

TRESOR Broker
RESTful interface

Technische Universität Berlin

Open Service Broker 1.0

TRESOR Open Service Broker

Main functionality

The TRESOR Open Service Broker is an information system realizing the following main functions:

- Managing service descriptions, clients, and providers
- Performing booking of services

Format of parameters

All described parameters can be sent using these methods:

- Managing service descriptions, clients, and providers
- Performing booking of services

Format of parameters

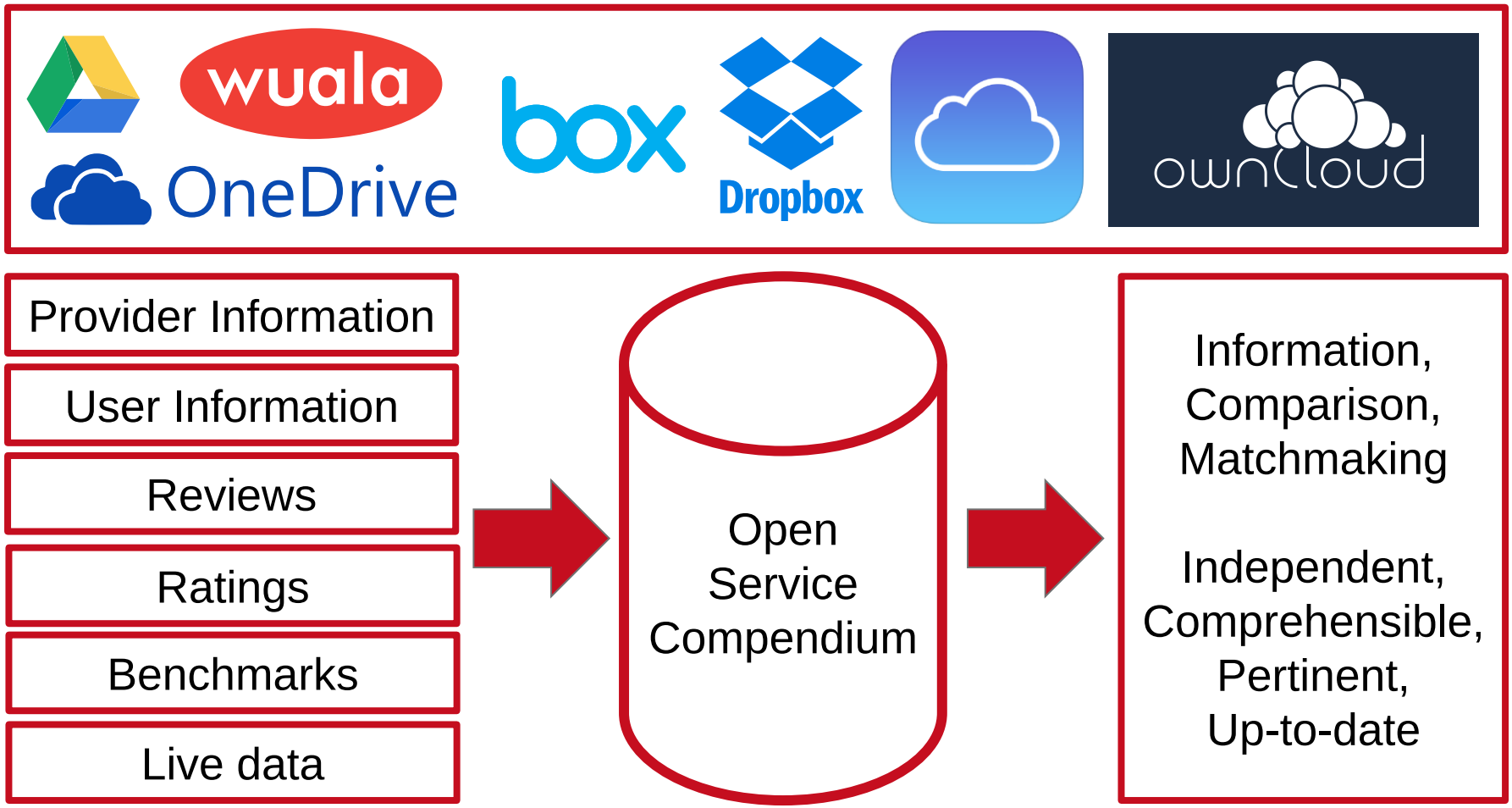
All described parameters can be sent using these methods:

- Sending parameters within the HTTP body, `multipart/form-data` encoded (on POST & PUT

TRESOR Broker
Further Functionality

- Search profiles
  - Simple DSL for matching service properties to predefined values, e.g., „Compatible browsers should include firefox“
  - Extension of search functionality currently in development
- Version management
  - Draft / approved services
  - Update workflow with automatic state update
  - Multi-version services
- Booking and provisioning
  - Currently only basic provisioning (single endpoint URL)
  - Possibility: complex deployment, e.g., using Slipstream

Vision: Open Service Compendium



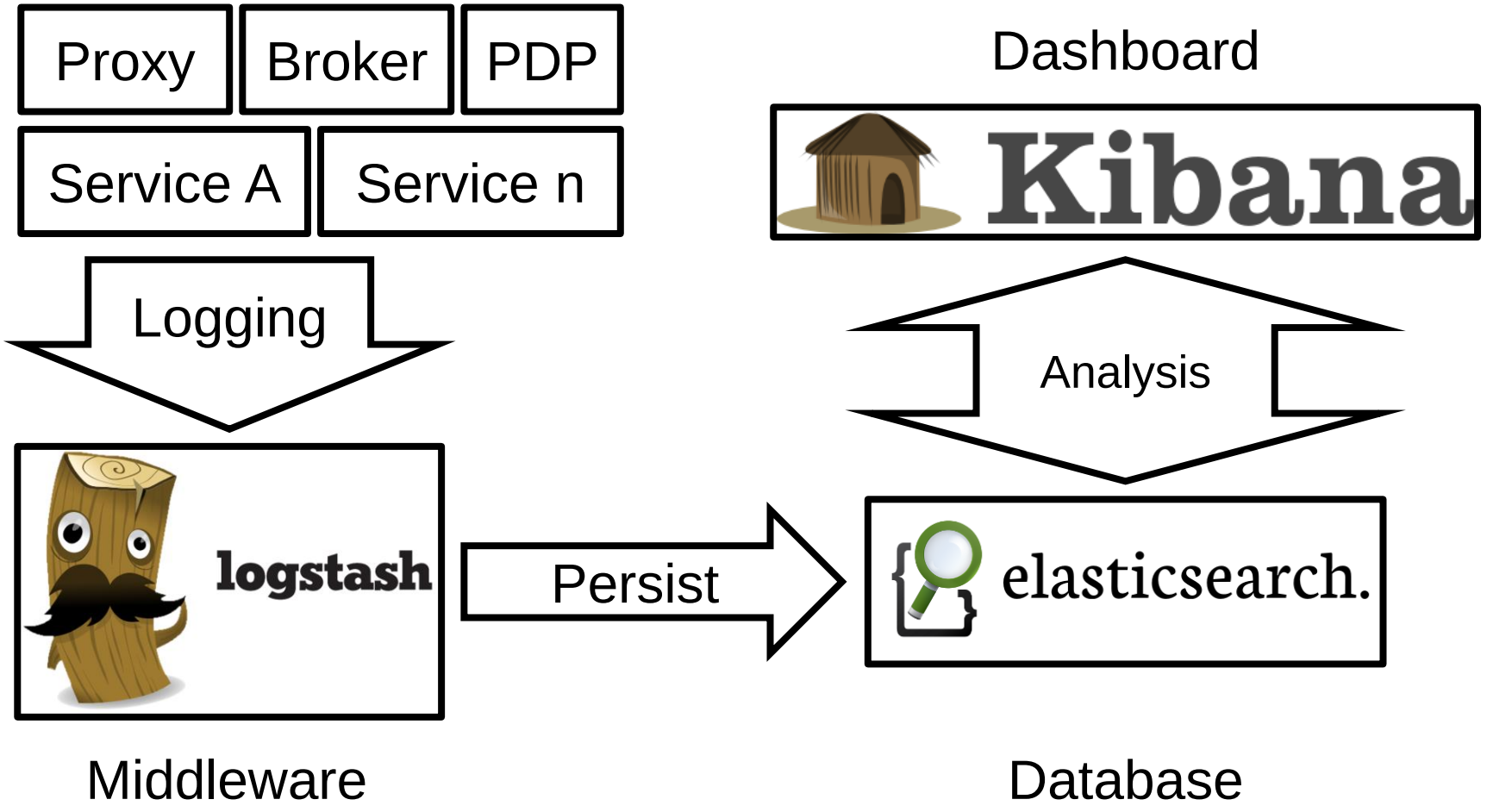
Technische Universität Berlin

DISTRIBUTED LOGGING

Distributed logging

- Logging of ALL actions through central instance
- Goals:
  - Trustworthiness
  - Legal compliance
  - Retrospective audits (even when changing services)
  - Monitoring of service quality
  - Simplified troubleshooting
  - Usability and consistency

Logging: ELK Stack



Logging: Demo-deployment

- `elasticsearch`
- `logstash`
- `esfilter`
- `kibana`

Log example


```
{
   "tresor-component": "Proxy",
   "logger": "Tresor::Backend::RelayingBackendHandler",
   "priority": "INFO",
   "category": "Routing",
   "message": "Successfully relayed POST http://... to http://...",
   "client-id": "92707293-e2e9-4d09-bad1-5c37bdfd0b09",
   "subject-id": "MMS\\ERB",
   "@timestamp": "2015-01-15T17:24:12.395+01:00",
   "@version": "1",
   "severity": "INFO",
   "host": "tresor-dev-proxy"
}
```

Demo-Dashboard

Technische Universität Berlin

Technische Universität Berlin

LIVE DEMO OF TRESOR ECOSYSTEM

Live Demo

- Running the ecosystem as fig deployment
- Broker
  - Logging into Broker through Proxy
  - Querying services
  - Creating a new service
  - Booking a service
- Administering policies for the new service
- Accessing a service with different policies
- Exploring log entries

Technische Universität Berlin

OUTLOOK & NEXT STEPS

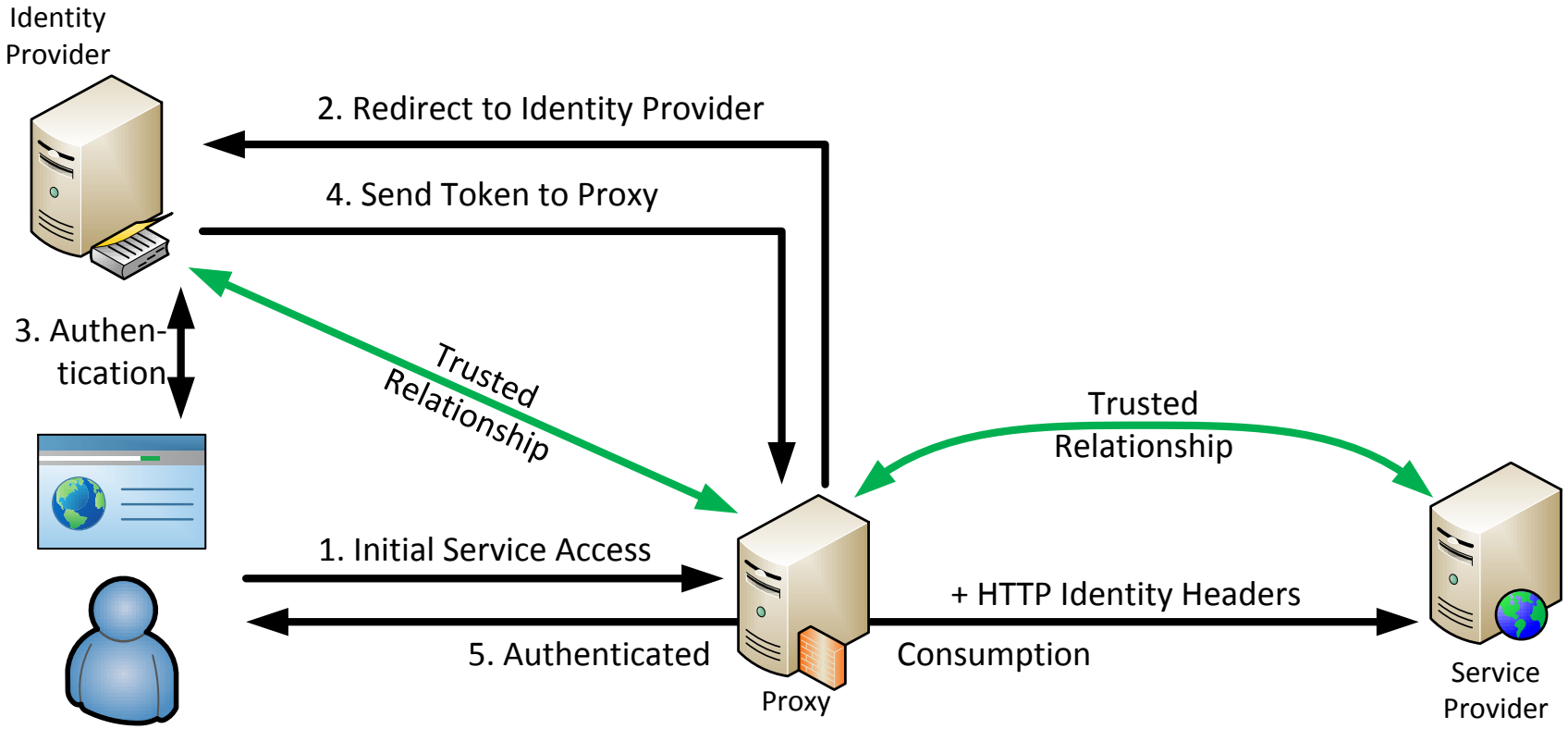
Authentication & Authorization

- Extending TRESOR notion of client/organization to CYCLONE federated cloud environments
  - E.g. SAML-token & XACML request structure
- Merging UvA infrastructure-focused security with SNET application-focused security
- Replacing Microsoft Federation / Identity with Open Source components

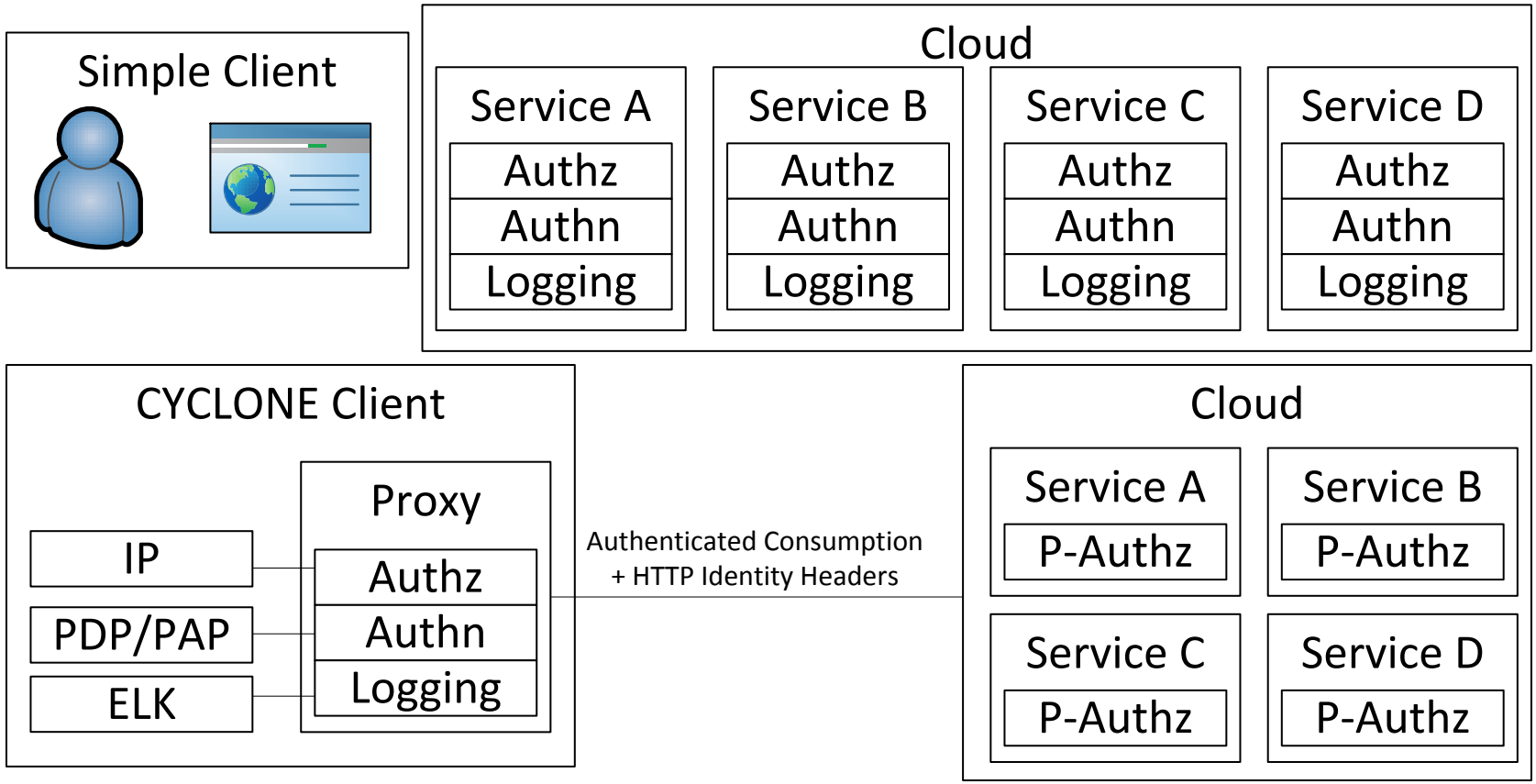
Proxy

- Deploying TCTP within use cases
- Integrating proxy functionality into cloud deployments and network management
- Using proxy as „trust anchor“ for secure authentication and authorization

Proxy as Federation Provider



Modularity in functional architecture



Broker

- Using broker information for chosing a deployment platform
- Solving open modelling challenges
  - Domain-specific criteria (e.g. IaaS platforms)
  - Handling variants in services
  - Raising intuitiveness of interfaces

Technische Universität Berlin

QUESTIONS / Q&A