

Entwicklung im Open Source Projekt dCache

HTW-Berlin

Big Data Projektstudium

Fritz Heiden

15.08.2017

Task #1

Update IoQueueManager allowing to Iterate the output of the 'mover ls' command.

Use Java8 stream API to iterate movers by client host name or IP address.

Was ist mover?

Was ist mover **Is**?

Task #1

Die Ausgabe des 'mover ls' Befehls soll nach einer Client IP oder eines Client Hostnames filterbar sein.

```
mover ls -client 127.0.0.1  
mover ls -client localhost
```

Die Implementation

```
@Argument(required = false, usage = "Limit output to mover with this job id.")
Integer id;

@Option(name = "queue", metaVar = "name", usage = "Limit output to this queue.")
String queueName;

@Option(name = "binary", usage = "Use binary output format.")
boolean isBinary;

@Option(name = "t", usage = "Sort output by last access time.")
boolean sortByTime;

@Option(name = "s", usage = "Sort output by transfer size.")
boolean sortBySize;

@Option(name = "r", usage = "Sort output in reverse order.")
boolean reverseSort;
```

```
@Argument(required = false, usage = "Limit output to mover with this job id.")
Integer id;

@Option(name = "queue", metaVar = "name", usage = "Limit output to this queue.")
String queueName;

@Option(name = "binary", usage = "Use binary output format.")
boolean isBinary;

@Option(name = "t", usage = "Sort output by last access time.")
boolean sortByTime;

@Option(name = "s", usage = "Sort output by transfer size.")
boolean sortBySize;

@Option(name = "r", usage = "Sort output in reverse order.")
boolean reverseSort;

@Option(name = "client", usage = "Filter output by client IP or hostname.")
String clientHost;
```



```
public Serializable call() {  
  
    StringBuilder sb = new StringBuilder();  
  
    queues.stream().forEach(q -> {  
        q.getJobs()  
            .sorted()  
            .forEach(j -> IoQueueManager.toMoverString(j, sb));  
    });  
  
    return sb.toString();  
}
```

```
public Serializable call() {  
  
    StringBuilder sb = new StringBuilder();  
  
    queues.stream().forEach(q -> {  
        q.getJobs()  
            .filter(clientIpFilter)  
            .sorted()  
            .forEach(j -> IoQueueManager.toMoverString(j, sb));  
    });  
  
    return sb.toString();  
}
```

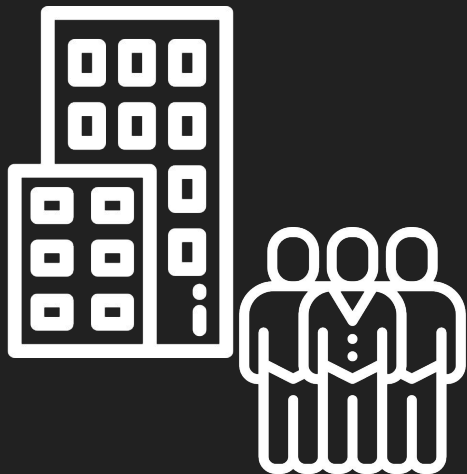
```
public Serializable call() {

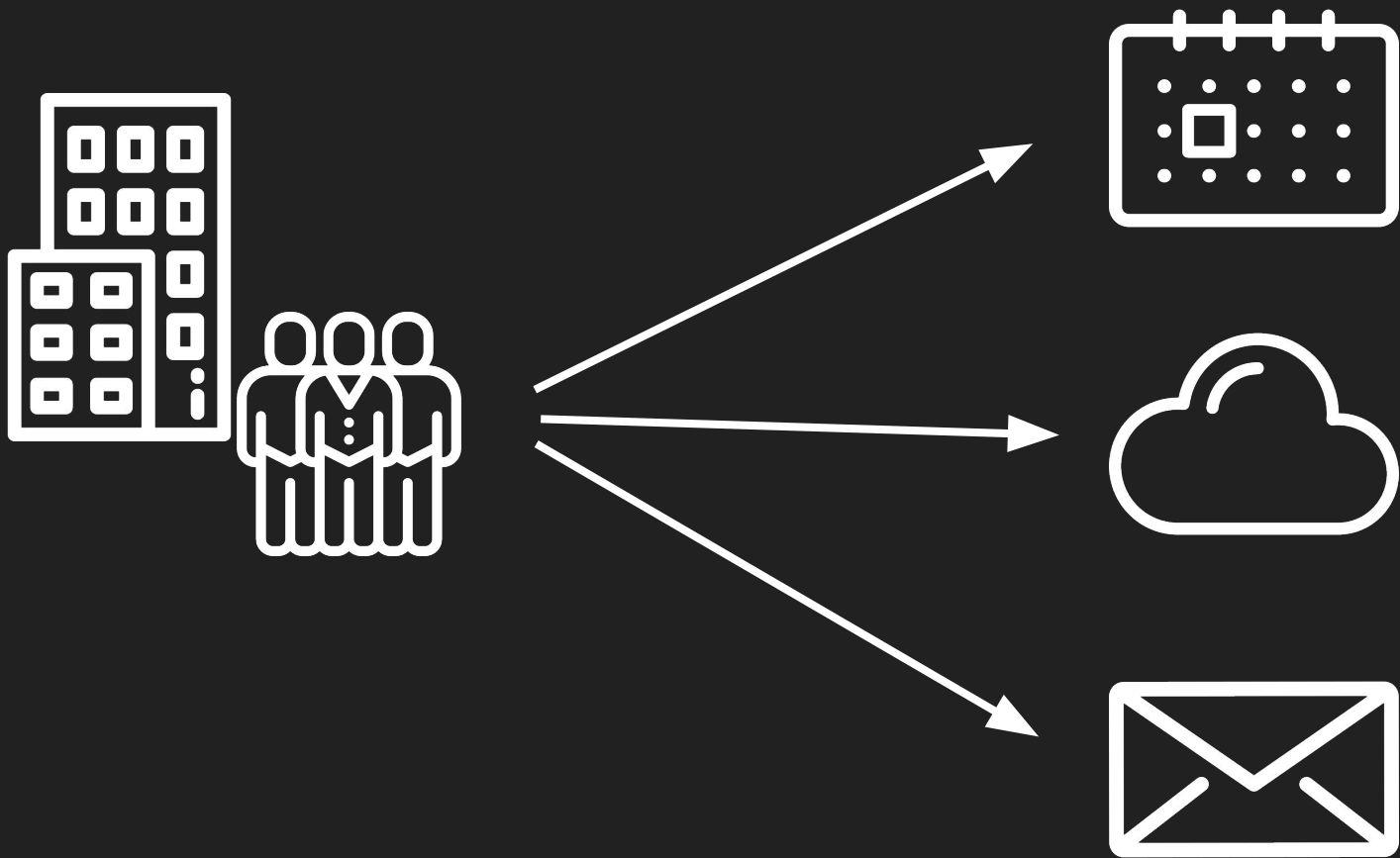
    Predicate<MoverRequestScheduler.PrioritizedRequest> clientIpFilter;
    if (clientHost != null) {
        InetAddress clientInetAddress;
        try {
            clientInetAddress = InetAddress.getByName(clientHost);
        } catch (UnknownHostException e) {
            throw new CommandException("Unknown host " + clientHost);
        }

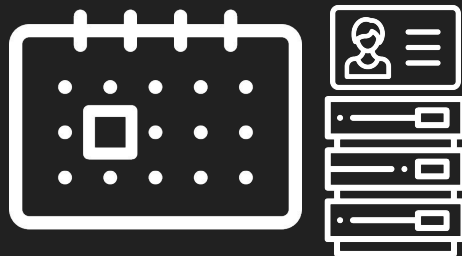
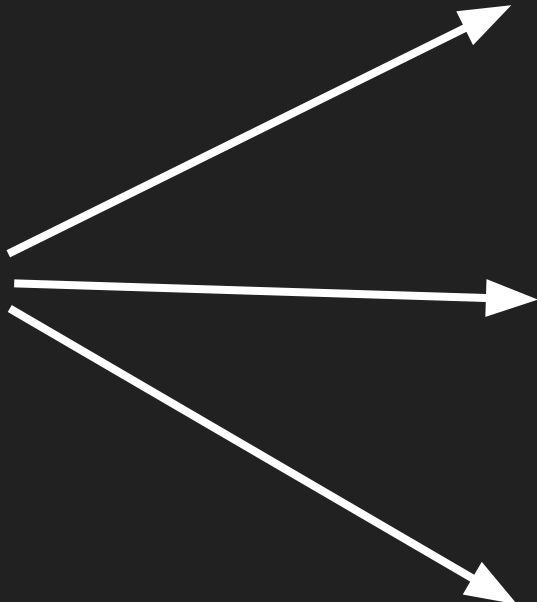
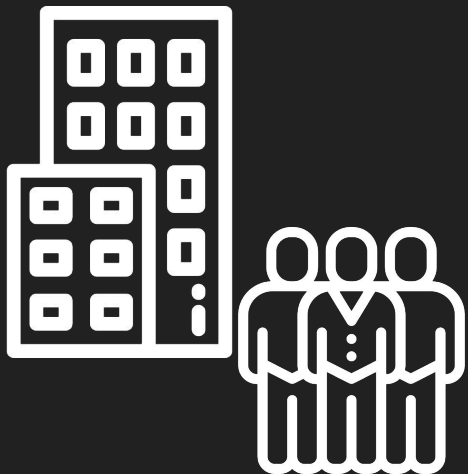
        clientIpFilter = prioritizedRequest -> {
            ProtocolInfo info = prioritizedRequest.getMover().getProtocolInfo();
            if (info instanceof IpProtocolInfo) {
                return ((IpProtocolInfo) info).getSocketAddress()
                    .getAddress().equals(clientInetAddress);
            } else {
                return false;
            }
        };
    } else {
        clientIpFilter = prioritizedRequest -> true;
    }
}
```

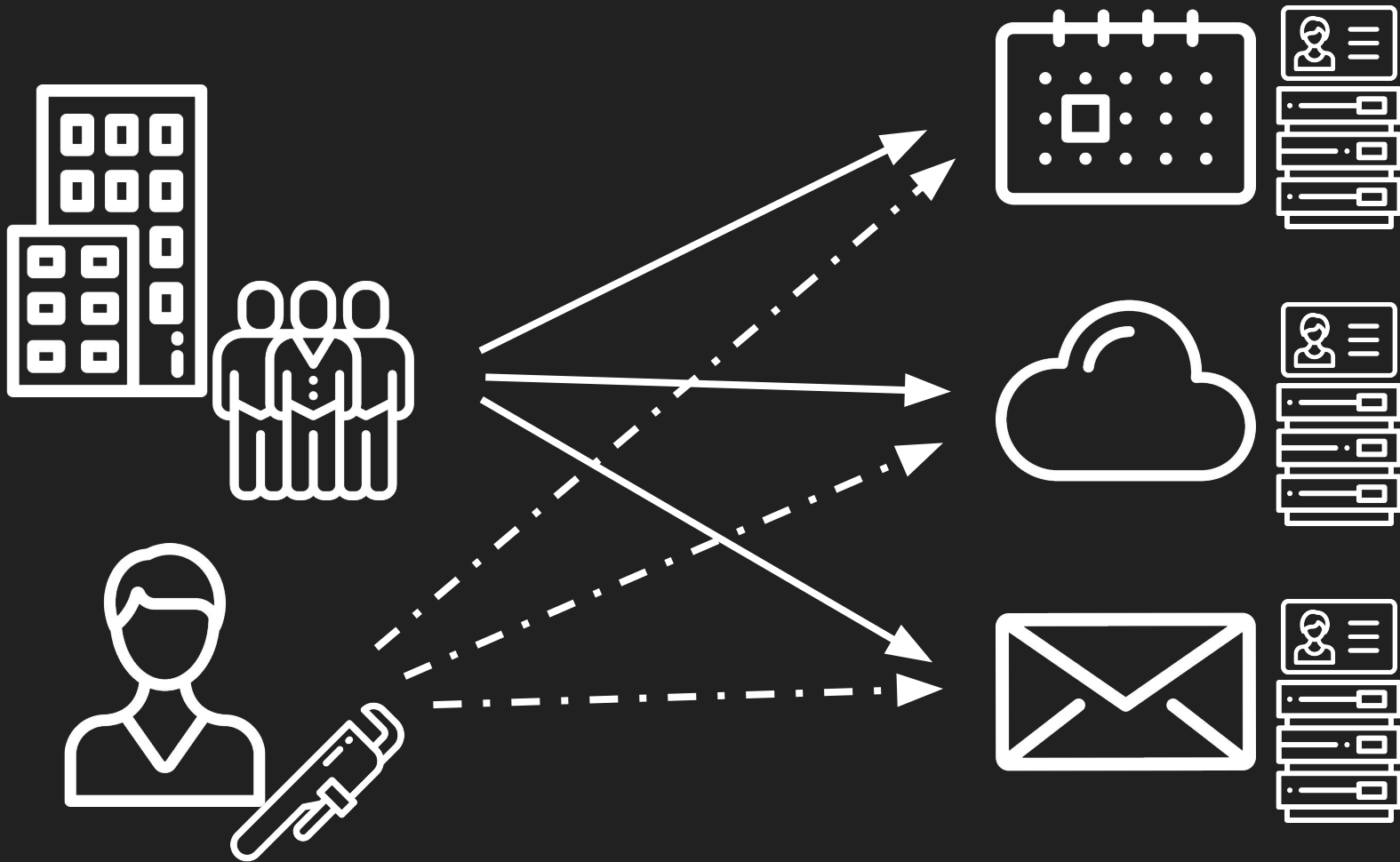
Wozu das Ganze?

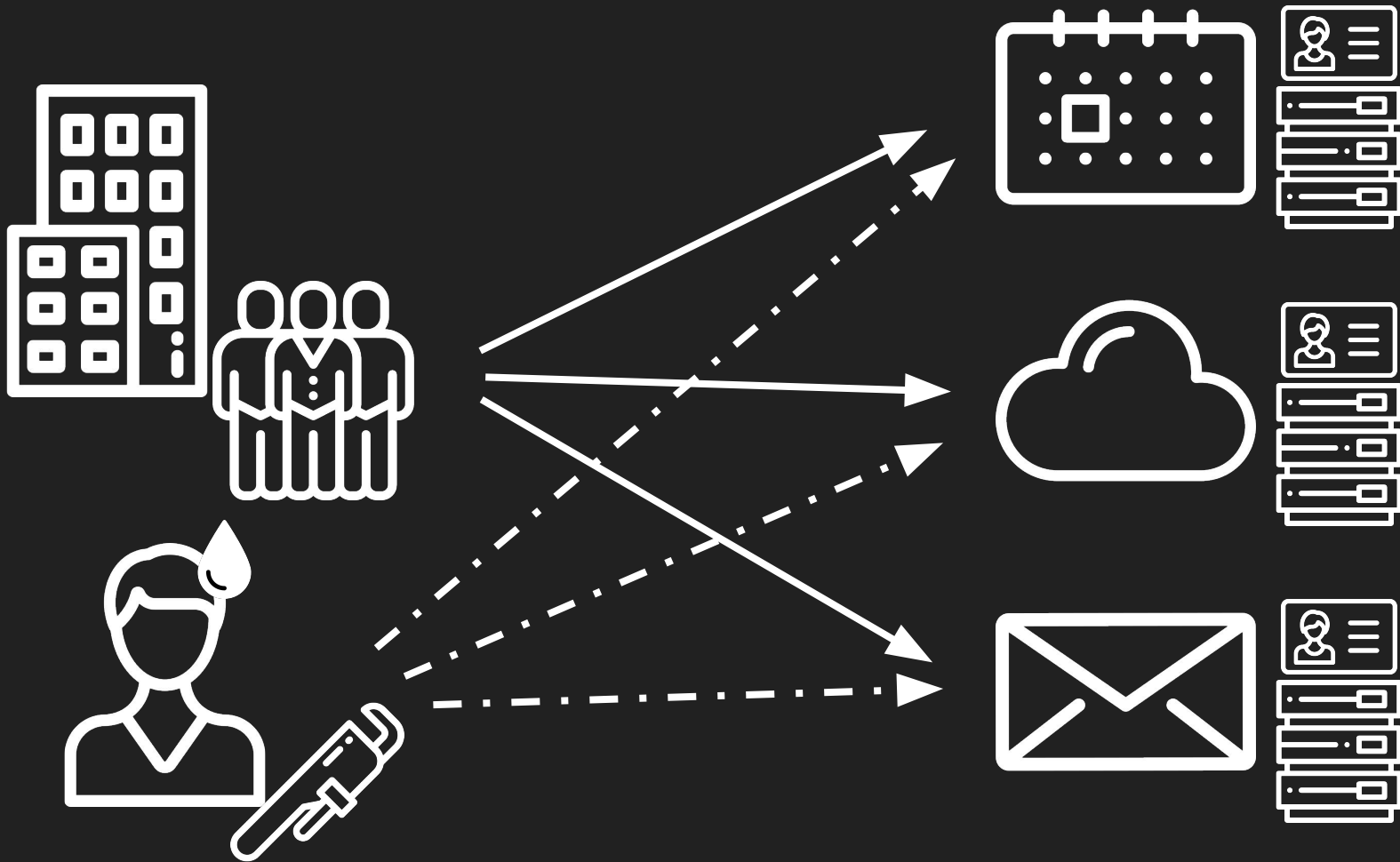
Security Assertion Markup Language 2.0 (SAML 2.0)

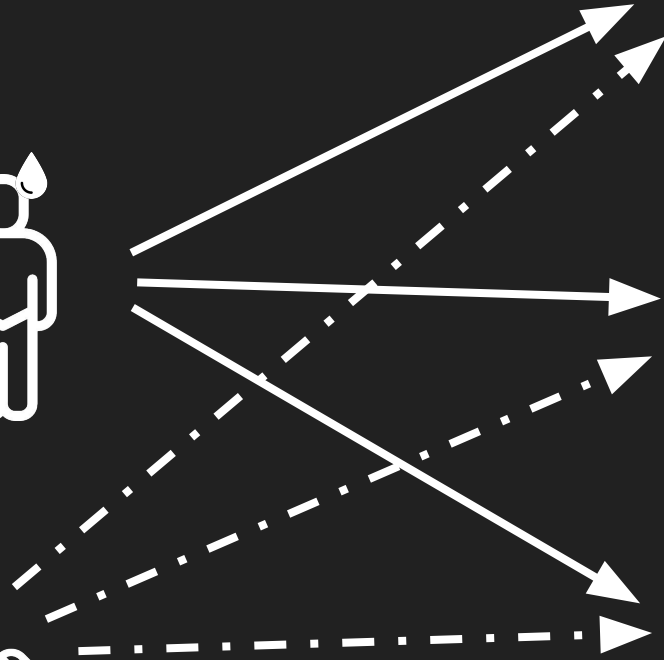
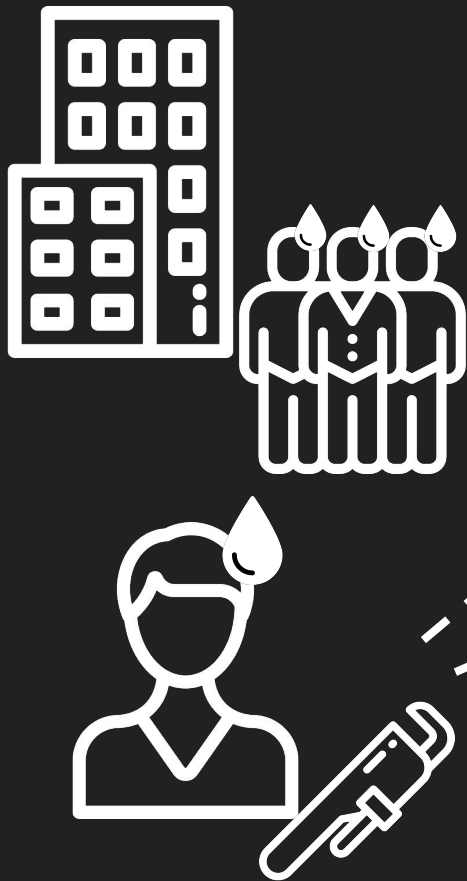


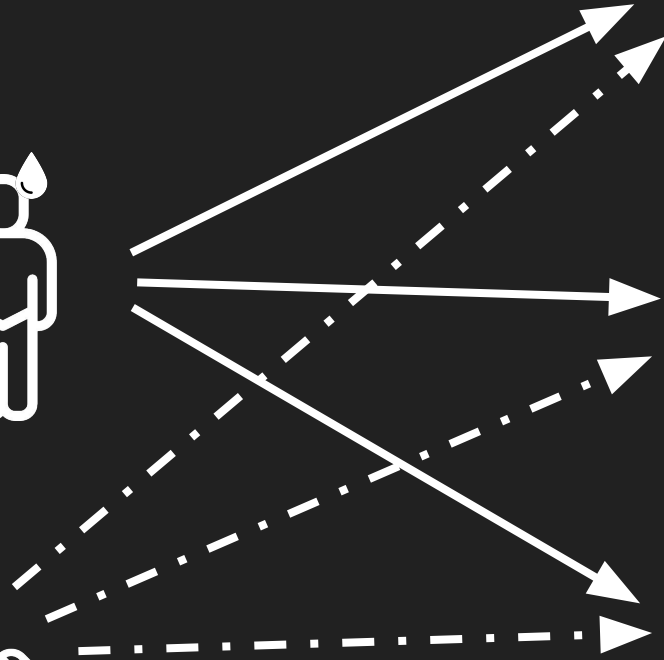
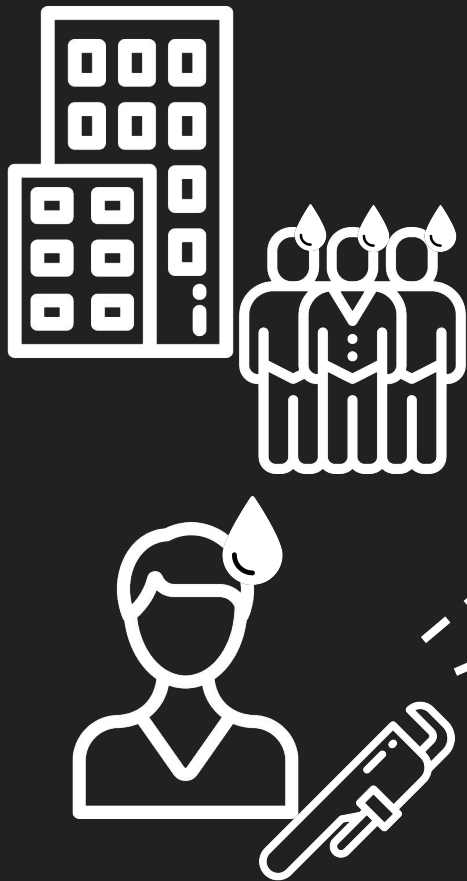


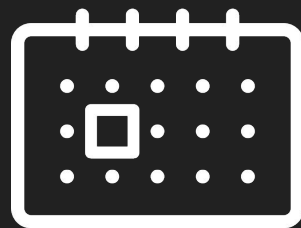
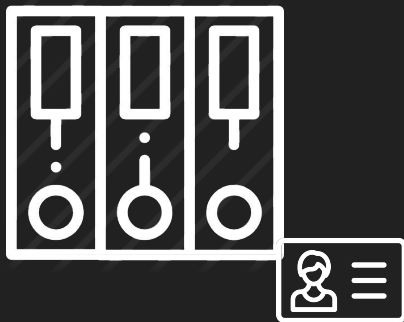
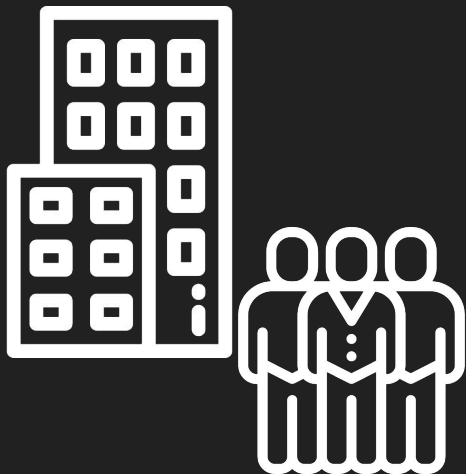


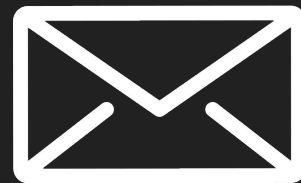
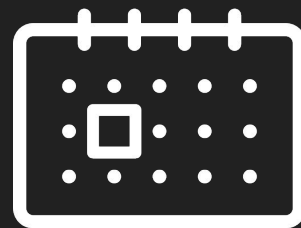
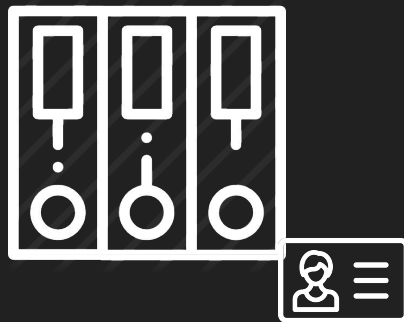
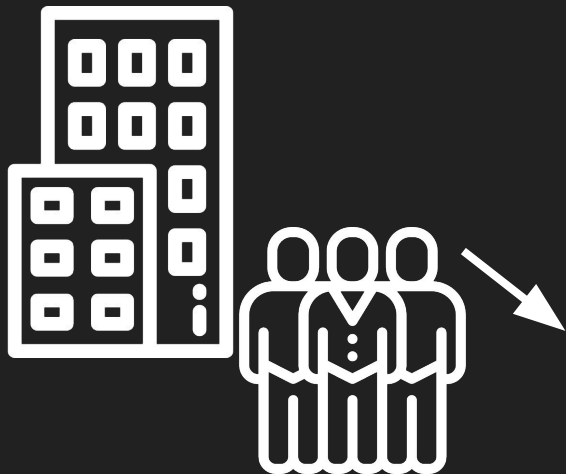




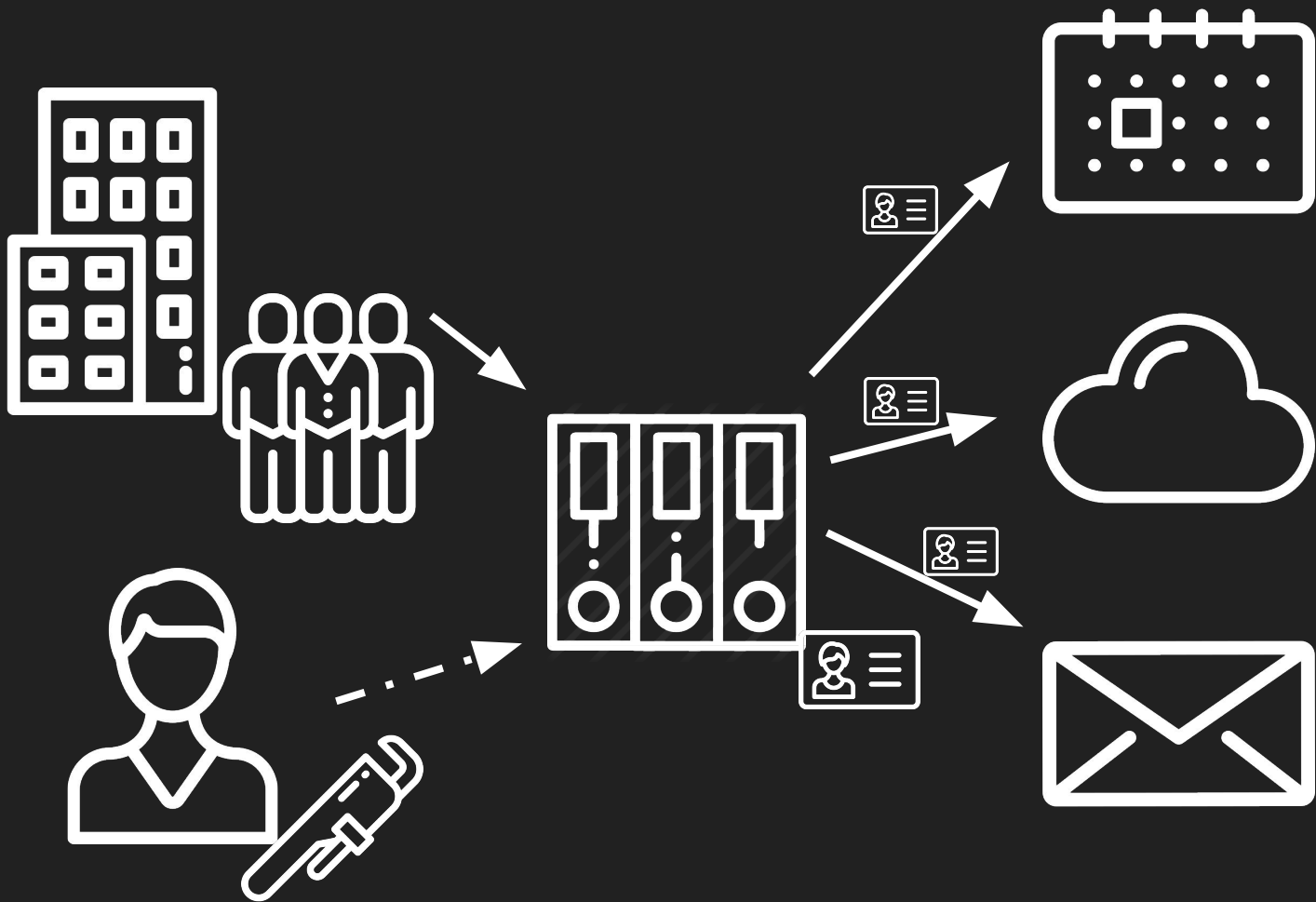


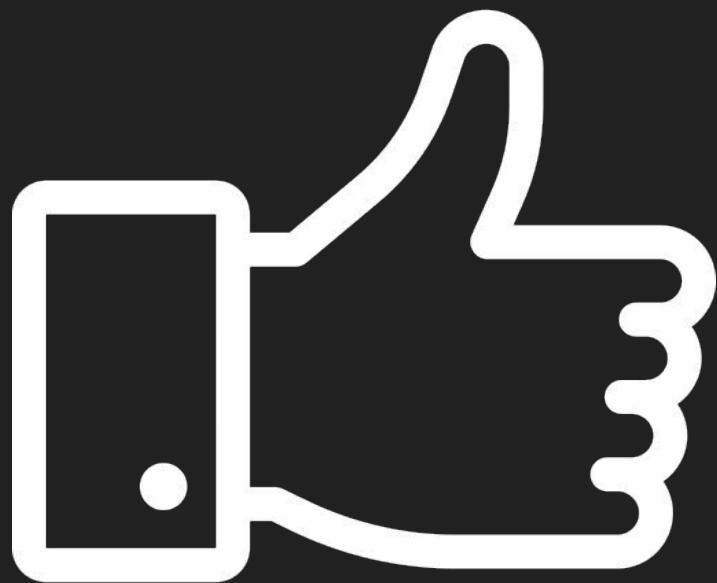




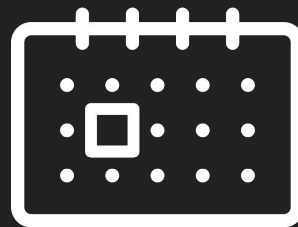
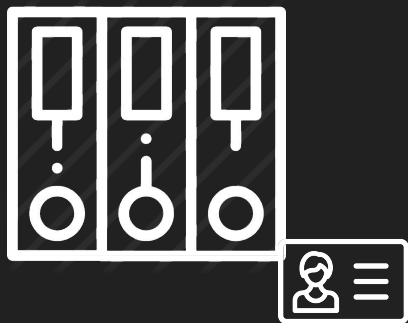


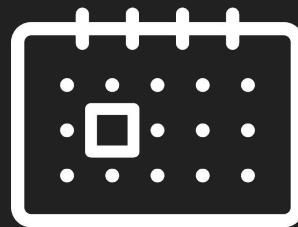
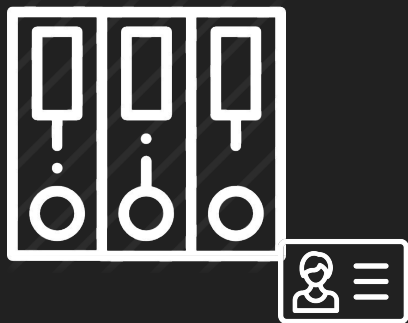


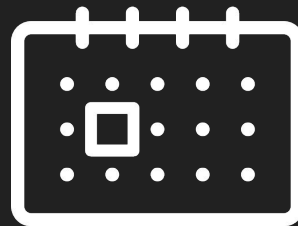
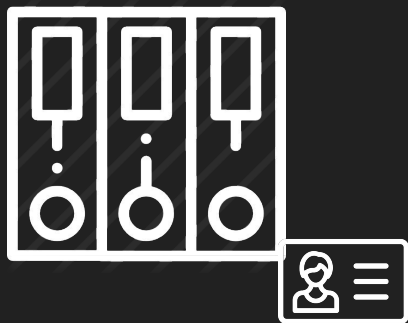


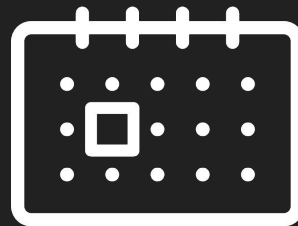
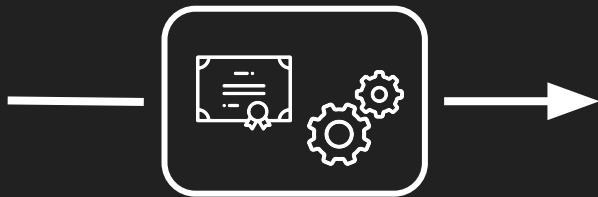
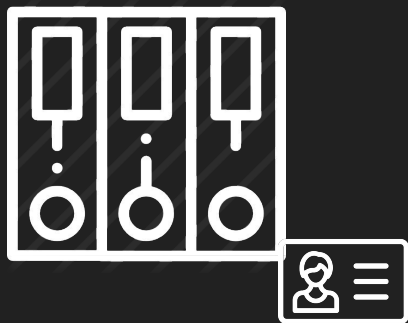


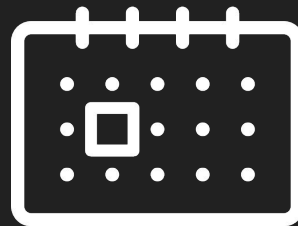
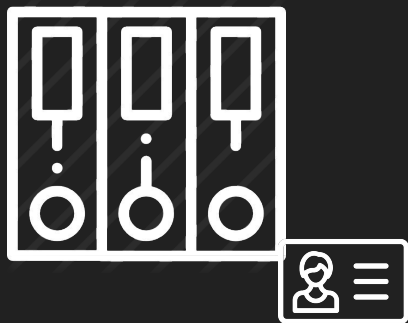
Das ist also SAML!











Wie sieht so ein Token
aus?


```
<saml:Assertion>
  <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
  <ds:Signature
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    [ ... ]
  </ds:Signature>
  <saml:Subject>
    [ ... ]
  </saml:Subject>
  <saml:Conditions
    NotBefore="2004-12-05T09:17:05"
    NotOnOrAfter="2004-12-05T09:27:05">
    [ ... ]
  </saml:Conditions>
  <saml:AuthnStatement
    AuthnInstant="2004-12-05T09:22:00"
    SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
    [ ... ]
  </saml:AuthnStatement>
  <saml:AttributeStatement>
    [ ... ]
  </saml:AttributeStatement>
</saml:Assertion>
```

Fragen?

Quellen (1)

Centrify:

- Introduction to SAML:
<https://www.centrify.com/resources/introduction-to-saml> [14.09.2017]

Tech Nugget:

- SAML Explained
<http://tech-nugget.blogspot.de/2013/09/saml-explained.html> [14.09.2017]

Quellen (2)

Wikipedia:

- Security Assertion Markup Language:
https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
[14.09.21017]
- SAML 2.0:
https://en.wikipedia.org/wiki/SAML_2.0 [14.09.2017]

Danke.