Zusammenfassung Future IT-Infrastructure FS2018

Alex Neher

September 25, 2018

Inhalt

Abbildungsverzeichnis

1 Netzwerk-Aspekte

1.1 VPN

Ein VPN (Virtual Private Network) erlaubt es einem Benutzer, sich in ein Netzwerk 'einzuklinken', selbst wenn er physisch nicht am Standort des Netzwerks ist.

1.1.1 Szenarien

Es gibt verschiedenste Szenarien, in welchen ein VPN nützlich oder vonnöten ist. Einige davon sind z.B:

Remote Access: Management eines Kundennetzwerks vom (Home-)Office aus, Zugriff auf HSLU-Ressourcen von zuhause. US-Netflix in der Schweiz schauen

Site-to-Site VPN: Verbindung zweier Netzwerke über einen verschlüsselten Tunnel. Von der Azure-Cloud ins EnterpriseLab, Verbindung zwischen zwei fixen IPs.

1.1.2 Technologien

Es gibt verschiedene Technologien, wie ein VPN aufgebaut sind. die zwei wichtigsten sind:

SSL-VPN Wird vor allem für **Remote Access** verwendet, da es mit einem relativ einfachen Setup verbunden ist. SSL-VPN läuft ausschliesslich über **Port 443 (HTTPS)**

IPSec-VPN Dieses Protokoll wurde ursprünglich exklusiv für IPv6 entwickelt, ist nun aber auch auf IPv4 portiert worden. Es kann, wie das SSL-VPN ebenfalls für Remote Access eingerichtet werden, ist jedoch auch nützlich für Site-to-Site VPNs. IPSec-VPN wird als **wichtigstes VPN-Protokoll** heutzutage angeschaut.

Des weiteren gibt es noch diese zwei, veralteten und unsicheren Protokolle, die jedoch immer noch eingesetzt werden.

PPTP (Point-to-Point-Tunneling Protocol): Basiert auf PPP [1], ist jedoch veraltet und unsicher.

L2TP (Layer 2 Tunneling Protocol): Ist unverschlüsselt und vertraut darauf, dass wichtige Daten verschlüsselt werden, bevor sie getunnelt werden. Dementsprechend veraltet und unsicher.

[1] Point-to-Pont-Protocol

1.1.3 IPSec

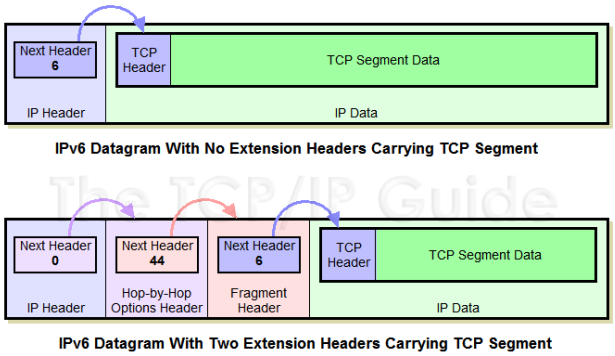


Abb. 1.1: IPv6 unterstützt optionale Headers, die zwischen Payload und Header eingeschoben werden.

Da IPSec ursprünglich für IPv6 entwickelt wurde, unterstützt es dessen Konzept der **Header Extensions** (Abbildung 1.1). IPv4 unterstützt jedoch *keine* Header-Extensions! Bei IPv4 werden diese IPSec-Header einfach zwischen dem IP-Header und dem TCP/UDP-Header eingefügt.

IPSec unterstützt die Verwendung des **AH** und/oder **ESP-Headers**.

AH (Authentication Header): Die Authentizität des Datenursprungs ist sichergestellt.

ESP (Encapsulating Security Payload): Die Daten sind verschlüsselt

Die eingefügten Header beinhalten eingentlich nur eine **Sequenznummer** und einen **Index auf eine SA** (Security Association). Zudem wird das gesamte Paket noch mit einem **Hashwert** versehen, was jedoch dank NAT zu Problemen kommen kann, da das NAT die Authentizitäts-Garantie des Authentication Headers versaut.

Security Association
Jedes IPSec Endgerät kann **beliebig viele** SA speichern. Eine SA ist grundsätzlich nur eine **Datenstruktur**, die (unter anderem) folgende Felder Informationen enthält:

Authentifationsverfahren: Modi und Schlüssel falls AH verwendet wird.

Verschlüsselungsverfahren: Modi und Schlüssel, falls ESP verwendet wird.

Lebensdauer der SA und Schlüssel: Wie oft muss der VPN-Tunnel wiederhergestellt werden.

IP-Adresse der End-Netzwerk Gateways

Beim Verbindungsaufwand wird diese SA aufgebaut. Dazu wird das **ISAKMP (Internet Security Association Key Management Protocol)** verwendet.

ISAKMP
Das ISAKMP Protokoll besteht aus zwei Phasen:

Phase 1: Ein **gemeinsamer Schlüssel** wird mit einem **erweiterten Diffie-Hellman-Verfahren** ausgehandelt (Bürgler lässt grüssen). Anschliessend werden die **Verschlüsselungs- und Hash-Protokolle ausgehandelt (nun in verschlüsselter Kommunikation)**. Dabei gibt es zwei verschiedene Modi: Im Main-Modus einigen sich beide Parteien auf die verwendeten Protokolle, während im Aggressiven Modus der Initiator 'seine' Protokolle vorgibt. Die Partner authentisieren sich via PSK, ihre Digitale Signatur, RSA oder El-Gamal.

Phase 2: Die SA wird nun aufgebaut und die weiteren Parameter für die IPSec-Verschlüsselung und den Tunnel werden ausgetauscht.

Transport- vs. Tunnel-Mode

Diese vorhin genannten weiteren Parameter sind z.B. ob der **Transport-** oder det **Tunnel-Modus** verwendet werden und ob **NAT-Traversal**[2] erlaubt sein sollte.

Bei IPSec ist per Default der **Tunnel-Mode** eingestellt. In diesem Modus wird das **gesamte IP-Paket verschlüsselt**. IPSec nimmt es, verschlüsselt es, packt seinen IPSec Header davor und schickt es durch den Tunnel. Die allfälligen AH- und ESP-Header werden zwischen dem 'alten' IP-Header des zu verschlüsselnden Pakets und dem 'neuen' IP-Header des IPSec-Protokolls gepackt (Abbildung 1.2 unten).

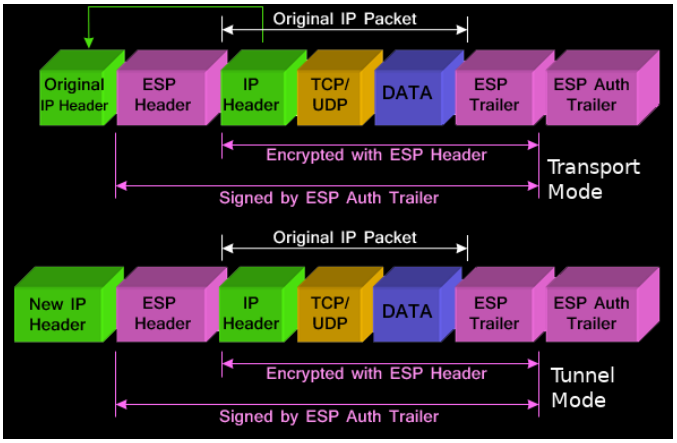


Abb. 1.2: Unterschied der Header zwischen dem Transport- und dem Tunnel-Mode

Beim **Transport-Mode**, der hauptsächlich bei **End-to-End-Verbindungen)** wie z.B. Client zu Server eingesetzt wird, wird das IP-Paket durch **AH und/oder ESP-Header** verschlüsselt. Der IP-Header des Pakets wird beim Transport-Mode an den Anfang geschoben, gefolgt vom ESP (und evtl. AH-Header).

Transport-Mode wird meist mit einem anderen tunneling-Protokoll wie z.B. GRE gekoppelt. So wird der original Payload zuerst von GRP umschlossen, bevor IPSec das neue, umschlossene Paket durch den Tunnel schickt.

Verbindungsprobleme mit IPSec

Es kann auch zu 'Show-Stopper' kommen beim Verbindungsaufbau mit IPSec, die verhindern, dass die Verbindung aufgebaut werden kann.

- Der Initiator ist im Aggressiv-Mode und der Partner unterstützt die verlangten Protokolle oder den Aggressiv-Mode nicht (z.B. Android)
- Bei Remote Access VPN müssen IP-Adressen und weitere Informationen dynamisch übermittelst werden, die evtl. nicht untereinander kompatibel sind.
- Es gibt kein gemeinsames Set von Verschlüsselungs- oder Hash Algorithmen der beiden Seiten, bzw. die unterstützten Schlüssellängen sind nicht komptabiel.

In aller Regel sind Site-To-Site VPNs aber trotzdem möglich, solange die verwendeten Algorithmen bekannt und kompatibel sind.

[2]Ermöglicht es, das AH-NAT-Problem zu umgehen, indem die Pakete in UDP-Pakete encapsulated werden, die anschliessend über Port 4500 versendet werden..

1.2 WAN-Technologien

1.2.1 Definition

WAN (Wide Area Network) sind **Netzwerk-Verbindungen** über **weite Distanzen**. Die Nutzen von WAN werden aufgeteilt in *internet-basierte* und *nicht-internet-basierte* Nutzen.

Internetbasierte Nutzen von WAN sind z.B. der Zugriff auf entfernte Ressoourcen über RDP oder Citrix, SSL-VPN oder Site-to-Site VPN via Internet.

Nicht internet-basierte Nutzen von WAN sind z.B. Dark Fiber (Eine direkte Point-to-Point Verbindung zwischen zwei Standorten, die nicht am Internet hängt. Dark Fiber hat eine Reichweite von max 50km), LAN-Interconnect (Die Vereinigung zweier LANs über WAN), Private Lines oder Business VPN.

1.2.2 Beurteilungskriterien

Die Qualität von WAN-Verbindungen kann anhand verschiedener Kriterien gemessen werden.

Geschwindigkeit: Kann die Geschwindigkeit, die der ISP verspricht auch garantiert werden?

Zuverlässigkeit: Wird die vereinbarte SLA zuverlässig eingehalten?

Kosten: Hat der Service ein vertretbares Kosten/Nutzen Verhältnis?

Redundanz: Garantiert der ISP eine Redundanz, falls eine Line unterbrochen wird?

Konfiguration & Management: Welche Möglichkeiten bietet der ISP?

Provider/ISP: Welche Reputation hat der ISP? Wo ist er (vgl. Datenschutz), wie steht er wirtschaftlich da?

Sicherheit: Wie schützt der ISP die Verbindung?

1.3 IPv6

IPv6 (Internet Protocol version 6) ist der Nachfolger des heute am weitesten verbreiteten Internet Protokoll IPv4. IPv4 wurde in den Anfangsjahren des Internets entwickelt und ware nie dazu gedacht, solche Mengen von Geräten zu managen, die heute am Internet hängen. Deshalb bietet es nur eine sehr begrenzte Anzahl von öffentlichen IP-Adressen an (4.3 Milliarden öffentliche IPs vs. ca. 11.2 Milliarden Geräte in 2018 [3])). Durch 'verschwenderische' Vergabe dieser IP-Adressen in den 90er Jahren und strukturelle Begebenheiten des Protokolls (subnetting als Beispiel), sind diese IPs heutzutage so gut wie erschöpft.

IPv6, das bereits 1998(!!) vom IETF standardisiert wurde, besteht aus einer 128bit langen hexadezimal-Zeichenfolge. Dadurch bietet das Protokoll **340 Sextilionen ($340 * 10^{36}$) Adressen**. Ausserdem bietet es weitere, nützliche Features wie eingebautes QoS, Mobile IP oder erweiterte automatische Konfigurationen der Netzwerkschnittstellen.

1.3.1 Adressierung

Eine IPv6-Adresse ist ein 128bit langer hexadezimal-String. Er wird in 8 Blöcke zu je 16bit unterteilt, die mit einem Doppelpunkt voneinander getrennt werden.

Da 128bit hexadezimal-Strings ein bisschen komplizierter sind wie 32bit dezimal-Nummern, gibt es einige Vereinfachungen und Guidelines, die das Arbeiten mit IPv6 vereinfachen sollen:

[3] Quelle: https://www.gartner.com/newsroom/id/3598917

- Falls ein Block mit Nullen startet, können diese weggelassen werden. Somit wird die Adresse `2001:0db8:0000:08d3:0000:8a2e:0070:7344` zu `2001:db8:0:8d3:0:8a2e:70:7344`
- Falls ein ganzer Block (oder mehrere aufeinanderfolgende) ausschliesslich aus Nullen besteht, so kann er ganz weggelassen werden. Das darf jedoch nur *einmal* pro Adresse gemacht werden. `2001:0db8:0000:08d3:0000:8a2e:0070:7344` kann also abgekürzt werden zu `2001:db8::8d3:0:8a2e:70:7344`.
- Die letzten 4 Bytes (32bit) einer Adresse dürfen auch in Dezimal angegeben werden. Dies ist vor allem hilfreich, wenn IPv4 Adressen zu IPv6 Adressen umgewandelt werden. So kann die IPv4 localhost Adresse `127.0.0.1`, die eigentlich zu `0:0:0:0:0:ffff:7f00:1` wird, kann zur besseren Verständlichkeit auch als `0:0:0:0:0:ffff:127.0.0.1` (oder abgekürzt `::ffff:127.0.0.1`) geschrieben werden.

IPv6 unterscheidet im Gegensatz zu IPv4 nicht zwischen public und private IP Adressen. Diese Tatsache eigentlich auch das NAT obsolet, dessen einzige Aufgabe es war, mehrere private IPs zu einer public IP zu mappen. Stattdessen wird eine Subnetzmaske definiert (standartmässig 64bit). Die ersten 64bit identifizeren das Netzwerk und die restlichen 64bit identifizieren den Host in diesem Netzwerk. `2001:0db8:85a3:08d3:1319:8a2e:0370:7347/64` bezeichnet also den Host/die NIC `1319:8a2e:0370:7347` im Netzwerk. Analog wie im IPv4 bezeichnet `2001:0db8:85a3:08d3::/64` das gesamte Netzwerk (wie `192.168.0.0` in IPv4).

IPv6 Adressen werden zwar nicht in private und public Adressen unterteilt, jedoch wird zwischen verschiedenen **Adressarten** unterschieden:

Link Local Unicast: Jede NIC/jeder Host, der IPv6-enabled ist, erhält automatisch eine Link Local Adresse. Sie wird **automatisch erstellt** und ist einzigartig für jeden Host und wird von dessen MAC-Adresse abgeleitet. Dieser Adresstyp wird **nicht geroutet**, soll heissen diese Adresse ist nur vom lokalen Subnetz aus anpingbar.

Mithilfe der Link Local Adresse wird z.B. Neighbour-Discovery gemacht. Ebenfalls wird normalerweise die Link Local Adresse des Routers als Gateway angegeben.

Unique Local Unicast: Wie bereits erwähnt, arbeitet IPv6 nicht mit private und public Adressen, was NAT eigentlich obsolet macht. Jedoch ist es manchmal trotzdem wünschenswert, ausschliesslich im lokalen Netz zu kommunizieren. Unique Local Unicasts werden **ausschliesslich im eigenen LAN geroutet**, nicht jedoch im Internet.

Global Unicast: Dies ist die öffentliche IP des Host. Sie wird **im LAN und im Internet geroutet**.

Multicast: Multicasts werden verschickt, wenn die Link Local Adresse eines Hosts noch nicht bekannt ist. Sie sind das **Equivalent des IPv4 Broadcasts**.

Site Local Unicast: Vorgänger des Unique Local Unicasts, veraltet.

1.4 Software Defined Networks

1.4.1 Momentaner State-of-the-Art

Momentan werden Netzwerke **statisch** aufgebaut. Es wird geplant, getestet und anschliessend aufgebaut. Jede Änderung unterliegt dem **Change-Management** und muss zuerst genehmigt werden. Dasselbe gilt für virtualisierte Netzwerke wie z.B. VMWare-Cluster etc.).

2 Identity Management

2.1 Geschichtliches

Als der Übergang von standalone-Systemen zu vernetzten Rechnersystemen stattfand, mussten Rechner plötzlich Informationen von anderen Rechnern im gleichen Netzwerk abfragen (wie z.B. IP- oder MAC-Adressen). Zudem mussten Benutzer in der Lage sein, Informationen/Dateien von anderen Rechnern abrufen und bearbeiten zu können.

In den 8er Jahren kam schlussendlich die Idee auf, eine weltweite Datenbank mit allen vernetzten Rechnern aufzusetzen:

2.1.1 X.500

sollte folgende Eigenschaften besitzen:

- Verteilte Informationen

  Jede Firma verwaltet seine Informationen selbst

  Jede Firma bestimmt, welche Informationen veröffentlicht werden, also von aussen zugänglich sind.

  Alle Firmen können auf veröffentliche Informationen von anderen zugreifen.

- Hierarchische Struktur mit objektorientierter Ausrichtung

  Vorwiegend geographisch aufgebaut.

  Hierarchie ist den konkreten Bedürfnissen anpassbar.

  Berechtigungen und Sicherheit sind auf jeder Hierarchiestufe definierbar.

- Hohe Verfügbarkeit

  Verteilte Datenbanken

  Replizierte Datenbestände

  Verteilte Administration

Es stellte sich jedoch recht schnell heraus, dass eine einzige, globale Datenbank schlicht und einfach zu komplex ist. Zudem wurde X.500 praktisch nicht genutzt, höchstens als E-Mail-Adressen-Verzeichnis. Daraus folgt, dass es fast keine Produkte auf dem Markt gab, was die gesamte Situation nicht wirklich verbesserte. Aufgrund dessen erachteten es Betriebssystem-Hersteller auch nicht für nötig, native Unterstützung dieser Norm zu implementieren. → **X.500 starb recht schnell wieder aus.**

2.2 Begriffe

DIT - Directory Information Tree

Der DIT wird verwendet, um Netzwerke darzustellen. Informationen werden hierarchisch dargestellt (siehe Abb. 2.1). Im FAlle von X.500 war die Hierarchie Country (C) → Organization (O) → Organization Unit (OU). OUs halten weitere OUs oder Objekte, die mit einem Common Name adressiert werden. Am Beispiel von Abb. 2.1 wäre ein Commmon Name

CN=Markus Waldmann, OU=Informatik, OU=I, O=HSLU, C=CH

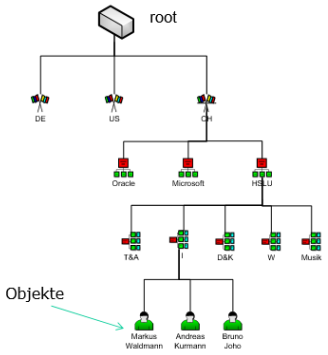
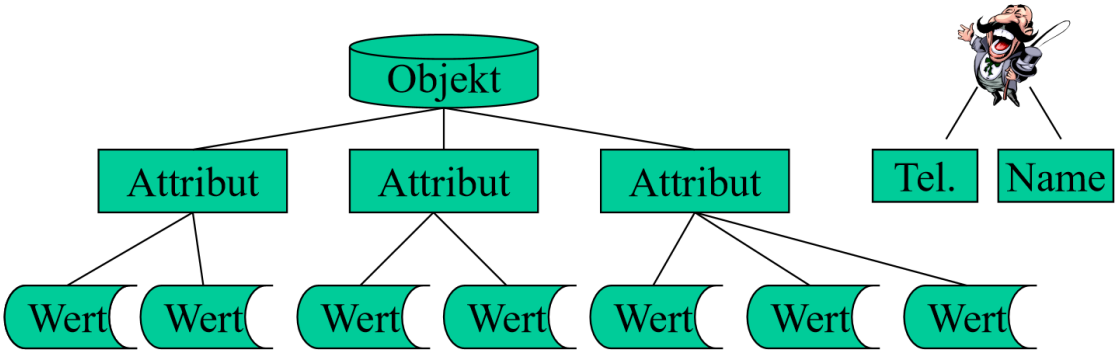


Abb. 2.1: Beispiel eines DIT

DIB - Directory Information Base

Die DIB ist eine Datenbank, in der alle Objekte des DIT gespeichert werden. Jedes Objekt hat verschiedene Attribute, die unterschiedliche Werte annehmen können.



Schema

eine Schema ist eine Ansammlung von Regeln, die Strukturen, Objekte und Datentypen beschreibt. Bei Bedarf kann ein Schema auch erweitert werden. Dank einem Schema ist eine Datenbank konsistent.

Beispiel von Regeln sind:

- Regeln zur Namenskonvention
- Regeln zur DIT-Struktur

  Wo können Objkete eingetragen werden?

  Welches sind zugelassene Super-Klassen?
- Regeln zu Objekt-Klassen

  Welhe Attribute kann und muss ein Objekt haben?
- Regeln zu Attributen

  Mögliche Datentpyen

  Mögliche Wertebereiche

ASN.1 - Abstract Syntax Notation
Wie vorhin bereits erwähnt, besteht ein X.500 System ausschiesslich aus Objekten. Diese Objekte müssen irgendwie beschreiben werden. ASN.1 ist eine Sprache zur Beschreibung eines solchen Objekts.

```
Name: Jan peter Schmidt
Geburtstag: 17.07.1957
...

PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
Name,
Titel [0] VisibleString,
DateOfBirth [1] Date,
(weitere Definitionen von Datentypen)
}
Name ::= [APPLICATION 1] IMPLICIT SEQUENCE {
Vorname VisibleString,
Mittelname VisibleString,
Nachname VisibleString
}
```

BER - Basic Encoding Rules
BER basiert auf ASN.1 und übersetzt komplexe Daten in einen Datenstrom.
Aus

```
Person    ::=  SET{
name      IA5String,
age       INTEGER
female    BOOLEAN
}

SET IA5String Maggie INTEGER 4 BOOLEAN TRUE
(Maggie ist eine 4 Jahre alte Frau)
```

wird

```
SET   IA5String     M  a  g  g  i  e  INTEGER     4  BOOLEAN     TRUE
31  14   16       06  77 65 71 71 73 69    02    01  04   01     01  FF
```

2.3 LDAP

LDAP steht für *Lightweight Directory Access Protocol* und ist prinzipiell nichts anderes wie X.500 mit dem TCP/IP Protokoll. Die Grundfunktionen von X.500 wurden beibehalten, jedoch lange nicht alle Funktionen.

Am Anfang wurden LDAP und X.500 parallel eingesetzt, indem ein Client übers Internet auf einen LDAP-Server zugreift, der die Anfrage schliesslich an einen X.500 Server weiterleitet (Abb. 2.2).

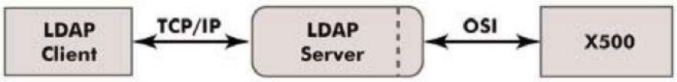


Abb. 2.2: X.500 mit einem vorgeschalteten LDAP-Server

Später wurden die Funktionen, die vom X.500 nicht ins LDAP übernommen wurden schlicht nicht mehr benutzt, und der X.500 Service konnte immer mehr weggelassen werden, was heute immer noch der Standard ist (Abb. 2.3).

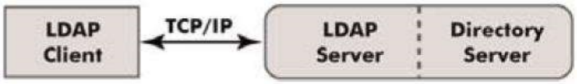


Abb. 2.3: LDAP-Server ohne X.500 System dahinter

Zusätzlich wurden neue Features ausschliesslich in LDAP implementiert, wie z.b. Sicherheitsanforderungen, Erweiterbarkeit, Zeichensätze oder auch die Möglichkeit fremde LDAP Server zu referenzieren.

Heute sind LDP Server Standard sowohl im UNIX wie auch im MS-Umfeld und immer mehr neue Anbieter (wie vCloud von VMWare) setzen auf LDAP. Jedoch ist die Interoperabilität zwischen einzelnen Produkten eher mangelhaft. Obwohl Interoperabilität zwischen verschiedenen Organisationen theoretisch möglich wäre, wird es praktisch nie eingesetzt.

2.4 Microsoft Active Directory

MS AD ist ein zentraler Ort für alle möglichen IT-Infrastruktur-Informationen wie

- Benutzerverwaltung
- File Sharing und Berechtigungen
- Drucker
- Digitale Zertifikate
- Mail- und Fax-Adressen
- Kontakte
- Workstation- und Server-Konfiguration

Dank dem AD müssen Applikationen nicht mehr einzeln auf 500 Workstations installiert und konfiguriert werden, sondern das kann alles zentral über sog. Group Policies erreicht werden. Das AD funktioniert auch als globales E-Mail Adressenverzeichnis.

MS AD unterstützt Standards wie LDAP, Kerberos (Authentifizierung), DNS, DHCP und RADIUS. Ausserdem bietet es ein ausgereiftes Administrationsmodell, welches eine Netzwerkweite Administration oder die auch die Delegation von einzelnen administrativen Aufgaben unterstützt. Das Active Directory kann auch in grossen Organisationen eingesetzt werden, da es auch mit mehreren Millionen Objekten problemlosläuft und dank einem organisationsweiten Schema werden auch Regeln für die Objekterstellung forciert.

Objekte können auf verschiedene Arten angesprochen werden:

LDAP-Qualifizierter Name: CN=MWaldmann, OU=I, DC=HSLU, DC=CH

UNC/URL Notation: hslu.ch/I/MWaldmann

UPN Notation: MWaldmann.I@hslu.ch

GUID: tawaldma

Die GUID ist insofern heikel, dass sie im gesamten AD gültig sind, das heisst "sekretariat" kann z.B. in einer grösseren Organisation mit verschiedenen Departementen nicht mehr verwendet werden, da der Name AD-weit eindeutig sein muss. → Eine AD-weite Namenskonvention muss definiert werden.

2.5 Identity Management

Entität = Etwas das existiert, etwas Seiendes.

Identität = Völlige Gleichheit, Übereinstimmung in allen Merkmalen

Eine digitale Identität ist die Teilmenge der Attribute einer Entität, welche diese Identität in einem bestimmten Kontext im Unterschied zu anderen Entitäten bestimmbar machen.

Eine Entität kann abhängig vom Kontext und den dadurch erforderlichen Attributen auch mehrere digitale Identitäten besitzen.

2.6 Laws of Identity

1. **User Control and Consent**
   Ich kann dem System, das meine Daten speichert vertrauen, indem ich selbst entscheiden kann, welche Identitäten und Daten ich wem preisgibt.

2. **Minimal disclosuder for a constrained use**
   Daten werden nur auf einer "need-to-know"-Basis gesammelt und nur so lange behalten wie unbedingt nötig. Dies aufgrund dessen, dass je weniger Daten gespeichert werden, desto weniger Daten können gestohlen werden.

3. **Justifiable parties**
   Mit wem werden welche Informationen geteilt?. Ich habe das Anrecht darauf zu wissen, wem meine Daten zu welchem Zweck zur Verfügung gestellt werden.

4. **Directed Identification**
   Es muss unterschieden werden zwischen "omnidirektional" und "unidirektionalen" Identifikatoren.

   Ein omnidirektionaler Identifikator identifiziert sich gegenüber jedem, der sich dafür interessiert wie z.B. die URL www.microsoft.com mit einem gültigen SSL Zertifikat. Jeder den

es interessiert, kann überprüfen, ob diese URL tatsächlich das ist, für was es sich ausgibt: Eine URL der Microsoft Corporation.

Der unidirektionale Identifikator wiederum identifiziert sich ausschliesslich gegenüber der fragenden Partei. Angenommen, eine Website fragt mich nach meinem Alter, dann werden diese Daten ausschliesslich an diese Website weitergeleitet und nirgendwo anders hin. Wenn ich 5min später auf eine andere Website geht, so kennt diese mein Alter nicht.

5. **Pluralism of Operators and Technologies**
   Es soll mehrere, verschiedene Identifikationssysteme geben.

   Es macht keinen Sinn, sich mit der AHV-Nummer beim Geschäfts-PC einloggen zu können. Denn der Staat hat Informationen über mich, die den Arbeitgeber vermutlich nichts angehen. Das heisst, für verschiedene Dinge sollten unterschiedliche Identifikationssysteme mit unterschiedlichen Features verwendet werden.

6. **Human Integration**
   Die bisherigen Laws of Identity befassen sich hauptsächlich mit System-System oder auch Benutzer-System Interaktion. Diese Kanäle sind bereits recht gut gesichert. Was ist aber mit der Benutzer-Benutzer Interaktion? Wie kann ich sicherstellen, dass der nigerianische Prinz, der mir sein ganzes Geld vermachen will auch tatsächlich der ist, für den er sich ausgibt und kein Betrüger?

7. **Consistent Experience Across Contexts**
   Wie bereits in Punkt 5 angesprochen muss es verschiedene Identifikationssysteme geben. Ich gebe auf dem E-Banking Portal nicht dieselben Informationen preis, die ich auf Facebook preisgebe. Es ist ein anderer Kontext und somit eine andere digitale Identität. Das Ziel ist es, diese Identitäten greifbar zu machen, z.B. mit einem Icon auf dem Desktop, das ich doppelklicken, bearbeiten oder auch löschen kann, wie es mir beliebt.

Zusammengefasst (und auf Englisch) kann man die Laws of Identity folgendermassen zusammenfassen:

> Putting all the laws together, we can see that the request, selection, and proffering of identity information must be done such that the channel between the parties is safe. The user experience must also prevent ambiguity in the user's consent, and understanding of the parties involved and their proposed uses. These options need to be consistent and clear. Consistency across contexts is required for this to be done in a way that communicates unambiguously with the human system components.
>
> As users, we need to see our various identities as part of an integrated world that nonetheless respects our need for independent contexts. (MSDN-Website für Laws of Identity)

2.6.1 IAM in Firmen

- Applikationsspezifische Applikationsverwaltung
- ID-Daten in verschiedenen ID-Stores abgelegt
- Benutzer muss sich mehrere Passwörter merken
- Meist keine einheitlichen Security-Policies
- Keine durchgehenden Berechtigungsvergabe-Prozesse
- Manuelles Erzeugen/Löschen von Accounts

- Kein zentrales Access Management
- Viele grössere Organisationen haben zusammengeschusterte Lösungen
- IAM-Lösungen langen auf dem Vormarsch

2.6.2 IAM im Internet

- Benutzer hat meist mehrere Identitäten

  Cloud

  Social Networks

  Shopping

  Online Games

  etc...

- Pro Identität nur ein Account möglich

  Wobei heute Login via FaceBook / Google vielerorts möglich ist

- Viele Benutzer greifen zu Notlösungen, um ihre Identitäten im Griff zu halten

  Überall gleiche Credentials

  Vergessene Credetials

  Passwörter aufschreiben

- Singe Sign On ist ein Bedürfnis

  Jedoch soll der Nutzer trotzdem noch die Kontrolle über seine Daten haben.

→ Federated Identity Management

2.6.3 Federated Identity Management

Federated Identify Management ermöglicht es Benutzer, sich einmal bei einem Identity Provider einzuloggen. Alle anderen Services vertrauen anschliessend diesem Identity Provider, so dass sich der Benutzer nicht mehr bei ihnen einloggen muss (→ Single Sign On).

Dank dem Federated Identity Management können nun system- und serviceübergreifend Authentifizierung und Autorisierung durchgeführt werden.

Federated Identity Management ist essentiell für Cloud-Computing. Die meisten grösseren Cloud-Anbieter unterstützen bereits eine Art von Federated Identity Managent (SAML2 / ADFS von Microsoft, OpenID Connect, Shibboleth)

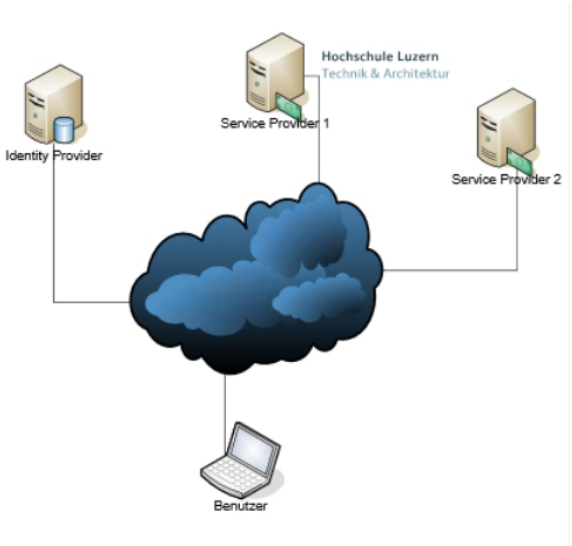


Abb. 2.4: Beispiel von Federated Identity Management

SAML 2
SAML (Security Assertion markup Language) ist eine organisationsübergreifende Auszeichnungssprache für den Austausch von Authentifizierungs- und Autorisierungsinformationen,die auf XML basiert.

Der Identity Provider ist die die Asserting Party (SAML Authority), welche die Identität eines Subjekts garantieren kann. Der Service Provider bietet einen Service an und vertraut dem Identity Provider.

SAML 2.0 definiert sechs Request-Response Protokolle,die über XML-Schemata definiert werden. Es gibt zum Beispiel den Authentication Request (Ein Service verlangt die Authentifizierung des Benutzers) oder den Assertion Query and Request (Der Identity Provider bestätigt, dass der Benutzer irgendwann irgendwie authentifiziert wurde).

Zudem werden über 6 sog. Bindings definiert, wie XML-Daten mit SAML übertragen werden wie z.B. über SOAP, HTTP Redirect oder HTTP POST.

SAML 2.0 wird in den letzten paar Jahren vermehrt in markttauglichen Produkten eingesetzt, wie z.B dem MS ADFS (Active Directory Federation Service)

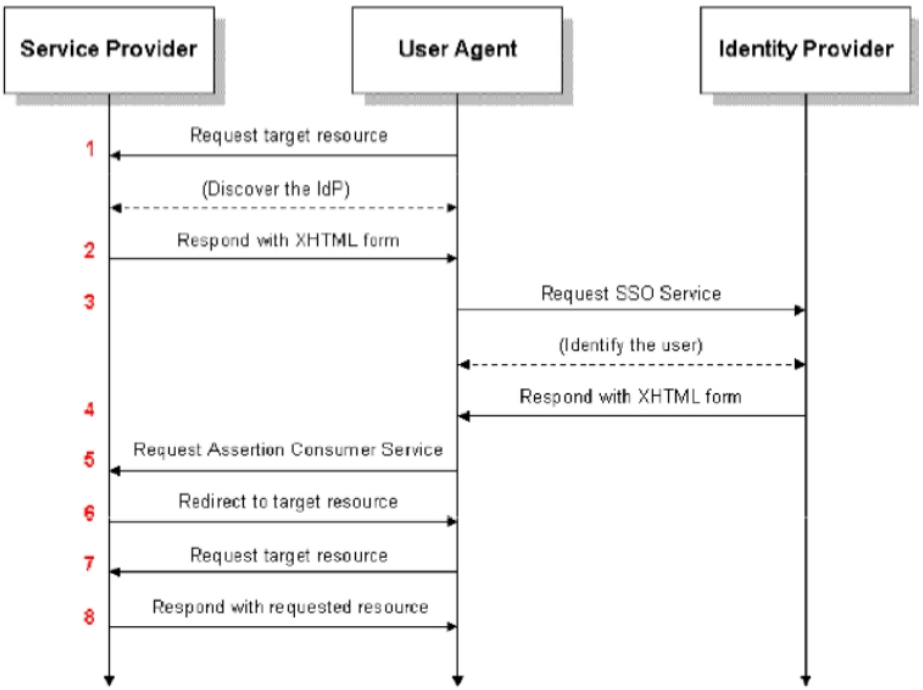


Abb. 2.5: Ablauf einer Authentisierung mit SSO Profil

Shibboleth
Shibboleth bietet ein Federated Identity Framework, welches vorwiegend im universitären Umfeld eingesetzt wird. Es ermöglicht Zugriffskontrolle, die auf einem Standard-Set von Attributen aufbaut, das jedoch bei Bedarf erweitert werden kann. Shibboleth ist zwar etwas Anderes als SAML 2.0, jedoch werden diese beiden Systeme immer kompatibler miteinander.

Die Authentifizierung über Shibboleth läuft folgendermassen ab:

1. Anfrage auf Ressource

2. Ressource redirected auf Discoveryservice (z.B. um auszuwählen zu welcher Hochschule ich gehöre)
3. User wählt die Hochschule aus
4. User wird zur Ressource redirected
5. Die Ressource leitet auf die Ressource der spezifischen Hochschule weiter
6. Dort muss sich der User mit den HS-Credentials anmelden
7. Der User wird auf die gewünschte Ressource weitergeleitet

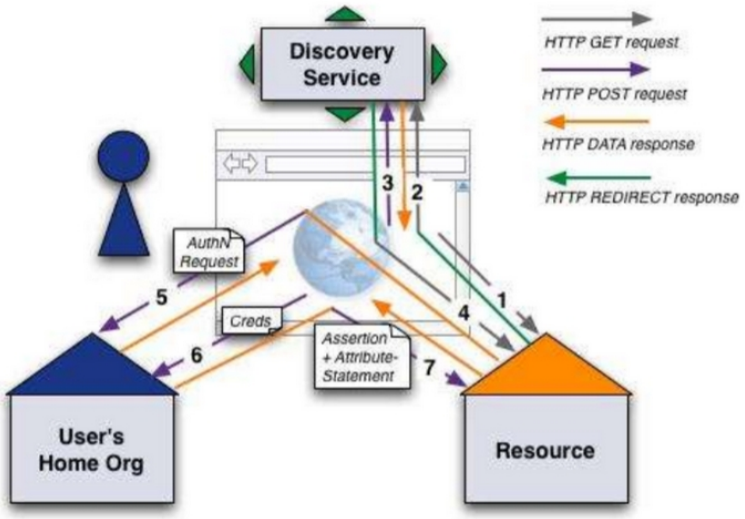


Abb. 2.6: Ablauf der Authentifizierung über Shibboleth

OpenID
OpenID ermöglicht die Authentifizierung über eine URL (z.B. portmal.clavid.com, www.clavid.com/portmann oder auch rportmann.startssl.com). Es gibt nur wenige Anbieter,die die Authentifizierung über OpenID zulassen und die Implementation hat noch einige Interoperabilitätsprobleme. Jedoch wäre dieser Standard Plattformunabhängig und stellt keinerlei Anforderungen an den Client.

OAuth []
OrganisationAuthentication ist ein open-source, tokenbasiertes Protokoll für die Autorisierung von APIs für sowohl Desktops wie auch Web- ud Mobile-Applikationen. Ein bekannteres Beispiel von OAuth ist die PKI-Karte. OAuth wird inzwischen von vielen grösseren Firmen intern verwendet und einige Cloud Anbieter haben es ebenfalls implementiert.

OpenID Conncect
OpenID Connect gehört nicht zu OpenID sondern zu OAuth. Es fügt dem Autorisierungsprotokoll noch einen zusätzlichen Authentifizierungslayer hinzu.

Digitale Signatur
Zumindest in der Schweiz ist die digitale Signatur der handschriftlichen rechtlich gleichgestellt. Jedoch gibt es noch einige Probleme. So gibt es in der Schweiz zur drei zugelassene CAs, viele Signatursoftwares überprüfen nicht alle PDFs, sondern nur PDF-A Formate. Zudem müssen einige Geschäftsprozesse angepasst werden und die Archivierung muss regelkonform sein. Deshalb gibt es erst einige wenige tatsächliche Anwendung der digitalen Signatur hier.

3 Cloud Resources

Fallstudie

4 Evalutation von Cloud-Services

Einige Aspekte die bei der Evaluation eines Cloud-Services berücksichtigt werden sollen:

- Kosten
- (Preis / Leistung)
- Funktionsumfang
- Vertragskonditionen (Wartung, Support, Ausstieg, usw.)
- Unterstützte Plattformen
- Sicherheit (Autorisierung, Datenschutz)
- Anpassungen des Service / Software
- Ruf der Firma
- Ausbau des RZ bezüglich Katastrophenschutz
- Wie oft gibt es einen neuen Release
- Einfachheit des Updateprozesses
- Performance
- Verfügbarkeit
- Skalierbarkei
- Wie viele andere Kunden gibt es
- Usability

4.1 Charakteristika eines Cloud-Services

On-Demand Self Service: Selbstzuweisung von Leistungen und Ressourcen aus der Cloud durch den Nutzer, die bei Bedarf bereitstehen.

Rapid Elasticity / Scalability: Funktionen und Ressourcen können schnell und dynamisch bereitgestellt werden, wenn möglich sogar automatisch. Aus Benutzersicht sind die Ressourcen "unlimitiert" und können jederzeit erweitert werden.

Broad Network Access: Die Services werden über ein internes oder externes Netzwerk zur Verfügung gestellt und können über standardisierte Schnittstellen auf unterschiedlichen Plattformen (wie Mobile oder Mac) genutzt werden.

Resource Pooling: Die Ressourcen des Providers werden nicht fest einem Benutzer zugeteilt, sondern alle verfügbaren Ressourcen werden in einem Pool gebündelt und dynamisch an die Benutzer vergeben, die es momentan brauchen.

Measured Service: Der Provider führt laufend QA durch und versucht, seinen Dienst stetig zu verbessern.

4.2 Merkmale und Service/Deployment Modelle

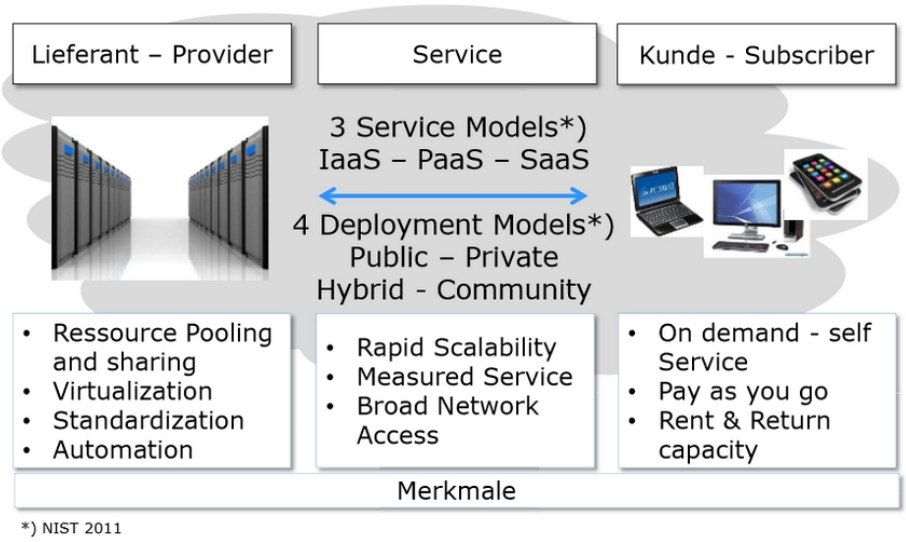


Abb. 4.1: Visualisierung der verschiedenen Service & Deployment Modellen

Deployment Model	**In der Regel bedeutet das**
Public Cloud	Für jedermann zugänglich. im Eigentum des Providers
Community Cloud	Nur für bestimmte Gruppen von Benutzern oder Unternehmen zugänglich. Im Eigentum der Nutzer/eines Providers
Private Cloud	Von einer einzigen Organisation genutzt. Kann im Eigentum dieser Organisation oder eines Providers sein.
Hybrid Cloud	Kombination aus verschiedenen Modellen.

Tabelle 1: Unterscheidung verschiedener Cloud-Modelle

Subkategorie	**Betreiber**	**Standort**	**Ressourcen-Sharing**
Intern	Interne IT	Nutzer	Keines
Outgesourced	Lieferant	Nutzer	Keines
Hosted	Lieferant	Lieferant	Innerhalb des RZ und Netzwerks
Virtual	Lieferant	Lieferant	Variabel

Tabelle 2: Subkateorien von Privaten Clouds

4.3 ERP- und E-Business-System

ERP System

- Aus mehreren Komponenten bestehendes integriertes Anwendungspaket
  (Integriert = Zieht Daten direkt aus der Datenbank, anstelle dass es selbst eine unterhält)
- Unterstützt die Abwicklung von Geschäftstransaktionen auf operativer Ebene
- Integriert in allen wesentlichen betrieblichen Funktionsbereichen
- Integration durch zentrale Datenbank
- Ermöglicht Abteilungsübergreifende Geschäftsprozesse

E-Business System
Ein ERP-System mit einigen zusätzlichen Funktionen wie z.B.

- Ermöglicht Betriebsübergreifende Prozesse
- Internet-Nutzung
- Zugang über Internet-Portale möglich

4.4 Evaluationsverfahren

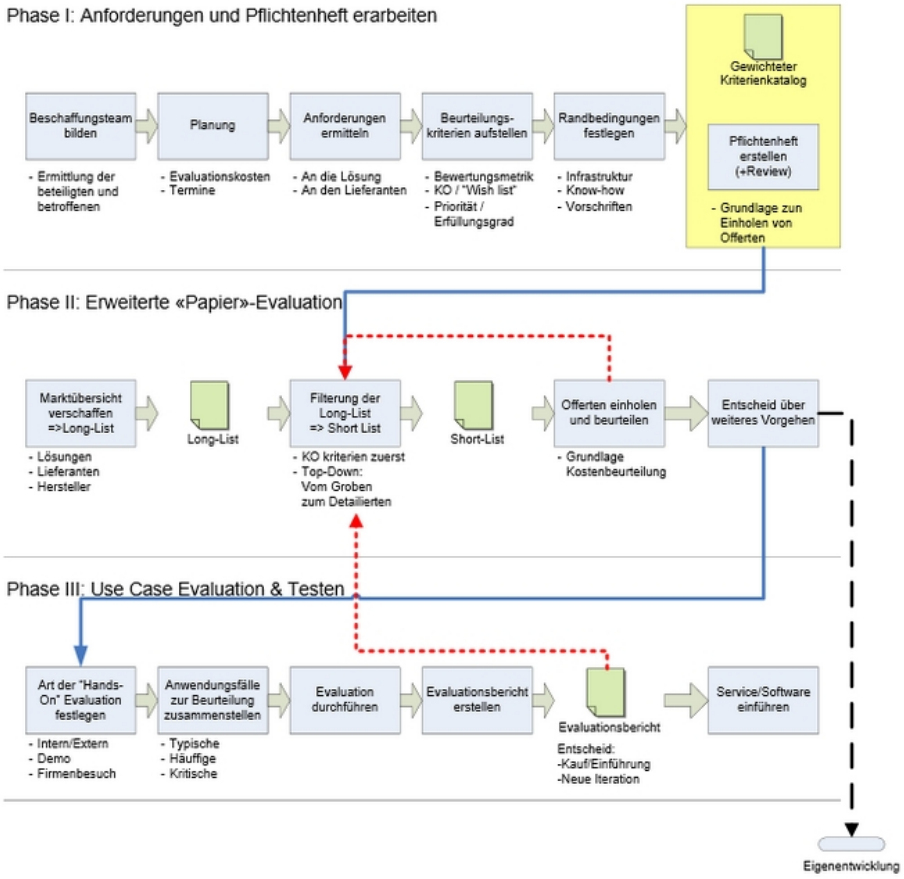


Abb. 4.2: Evaluationsvorgang bei der Auswahl eines Cloud-Services

4.4.1 Phase I - Anforderungen und Pflichtenheft erarbeiten

Beschaffungsteam bilden
Das Ziel dieses Arbeitsschrittes ist ziemlich selbsterklärend: Es soll ein Team gebildet werden, welches für die Evaluation und die Beschaffung des Services verantwortlich ist. Wichtig ist, dass die Endnutzer ebenfalls in den Prozess miteinbezogen werden.

Es sollten also Anforderungen für die gewünschten Personen definiert werden. Personen, die diese Anforderungen erfüllen können ins Team aufgenommen werden. Allenfalls muss noch ein externer Fachmann hinzugezogen werden.

Ebenfalls sollten alle involvierten Benutzer (also der Auftraggeber, die Steuerungsgruppe, der Projektleiter und die Endbenutzer) informiert werden.

Planung
Das Ziel dieses Arbeitsschrittes ist es, den Scope, das Vorgehen und das Budget definiert zu haben. Es sollte ebenfalls bereits eine Deadline definiert sein. Schlussendlich sollte ein genehmigter Kosten- und Terminplan stehen.

Die Planung beinhaltet folgende Schritte:

1. Ressourcen und Kapazitäten erfassen
2. Evaluationsvorgang planen
3. Kosten der Planungseinheiten abschätzen
4. Planung abnehmen lassen

Anforderungen ermitteln
Bevor man sich für einen Service entscheiden kann, sollte man vorher noch abklären, zu was der denn überhaupt fähig sein müsste. Es muss also ein Anforderungskatalog erstellt werden, in dem die Anforderungen an den Service, dessen Dienstleister oder Lieferanten festgelegt werden.

Um eine Anforderung festlegen zu können, muss sie zuerst definiert werden. Was ist eine Anforderung?

- Eine Forderung im Bezug auf etwas Bestimmtes

  z.B. Daten, Menen, Kosten, Performance etc.
- Diese Forderung steht einem direkten Nutzen gegenüber

  z.B. weniger Kontrolle nötig, mehr Automatisierung möglich etc.
- Der Nutzen muss quantifizierbar sein.

  "Wir wollen 15% unserer Arbeit automatisieren können."

Der Service wird in Muss- und in Kann-Anforderungen unterteilt. Die Muss-Anforderungen sind ein essenziell. Wenn ein Service aus der Liste eine Muss-Anforderung nicht erfüllt, so ist er raus.

Kann-Anforderungen sind "nice to have" und machen etwa 90% aller Anforderungen aus.

Anforderungen werden nach einer umgekehrten Pyramide geprüft. Wenn ein Service bereits durch die grobkörnige Prüfung rasselt, so kann er in den feinkörnigen Prüfungen gar nicht gut absschneiden. So kann Aufwand gespart werden, indem nicht bei jedem Service jede Anforderung bis ins Detail geprüft wird.

Ein Beispiel einer solchen umgekehrten Pyramide:

Grobkörnig: Der Service muss einen Webshop haben

Feinkörnig: Der Service muss einen Webshop mit integrierten Bewertungssystem und one-click Shopping haben

Wenn der Service keinen Webshop anbietet, so wird er auch keinen Webshop mit integriertem Bewertungssystem anbieten und man kann sich somit die Prüfung der weiteren Anforderungen getrost sparen.

Man kann das Vorgehen bei der Anforderungsermittlung also folgendermassen zusammenfassen:

1. Ist-Zustand erfassen

   Welche Prozesse und Ressourcen sind betroffen?

   Wie sieht unsere momentane IT-Infrastruktur aus?

2. Allgemeine und Softwarespezifische Checklisten konsultieren

   Fachliteratur, Internet etc.

   Damit keine wichtigen Punkte vergessen gehen

3. Workshops mit Business-Siete durchführen

4. Anforderungskatalog erstellen

   Nach Detailgrad strukturieren

   In "Muss" und "Kann" Anforderungen aufteilen.

Anforderungen an ERP-Services und -Software
Anforderungen an ERP-Systeme lassen sich grundsätzlich in 5 Teilgebiete aufteilen:

Functional Fit: Anforderungen an den Funktionsumfang; So wenige Funktionen wie möglich, aber so viele wie nötig.

Flexibility: Anforderungen an die Integrationsfähigkeit sowie den zu betreibenden Anpassungsaufwand.

Kann aufgeteilt werden in *Services & Software* und *Software-Pakete*
Services & Software

- Benutzerfreundlichkeit
- Skalierbarkeit
- Technologie (Stack/Sprache)
- Anpassungsfähigkeit
- Internationalisierung
- Schnittstelle

Softwarepakete

- Betriebssystemunabhängigkeit
- DB-Unabhängigkeit
- Upgrade/Update-Fähigkeit
- Architektur

Maturity: Anforderungen im Bezug auf die Ausgereiftheit eines Produktes oder die Erfahrung des Herstellers

Support: Anforderungen bezüglich Support-Leisungen des Herstellers

Continuity: Anforderungen bezüglich der technologischen und wirtschaflichen Stabilität des Herstellers

Beurteilungskriterien aufstellen
Nachdem die Anforderungen definiert wurden, müsste man auch noch wissen, wie man denn bewerten kann, wie gut eine Anforderung erfüllt oder eben nicht erfüllt wurde. Das Ziel dieses Arbeisschrittes ist es, schlussendlich für jede Funktion ein Rating zu haben (*Rating(Lösung)*). Nur so kann man effektiv verschiedene, scheinbar gleiche Lösungen miteinander vergleichen und schlussendlich abwägen, welche davon die Beste ist.

Für Muss-Anforderungen gestaltet sich das Beurteilungskriterium recht einfach: Ist die Anforderung vollauf erfüllt, ist gut; sonst fliegt die Lösung aus dem Katalog. Bei Kann-Anforderungen gestaltet sich das je nach dem ein bisschen schwieriger. Deshalb gibt es das **Prioritäts-Erfüllungsgrad Modell**.

Bei jeder Lösung wird für jede Kann-Anforderung der folgende Wert berechnet:

$$w_i = P_i * E_i$$

wobei P_i für die Priorität und E_i für den Erfüllungsgrad der Anforderung i steht. Das schlussendliche Rating der Lösung L ergibt sich dann aus

$$\sum_i w_i$$

Priorität P_i	**Erfüllungsgrad** E_i
0. Irrelevant	0. Nicht erfüllt
1. Geringer Nutzen/Nicht wichtig	1. Für nächste Version geplant
2. Mittlerer Nutzen/Kompromiss denkbar	2. Anpassung oder Workaraound erforderlich
3. Hoher Nutzen/Nur schwer verzichtbar	3. Erfüllt die Anforderung weitgehend
	4. Erfüllt die Anforderung voll

Randbedingungen festlegen
Zusätzlich muss festgelegt werden, welche Randbedingungen herrschen. Das können sein:

- Bestehende Systemlandschaft
- Vorhandene Hardware/Lieferanten
- Vorhandenes Know-How
- Gewohnheiten und Erfahrungen

Dabei geht man folgenderweise vor:

1. Fakten ermitteln, die nicht geändert werden sollten
2. Einfluss dieser Fakten auf den Anforderungskatalog erfassen und auswerten
3. Gegenfalls Anforderungen anpassen und/oder neue hinzufügen

Die Randbedingungen sollten nicht zu eng gefasst sein, da sonst der Lösungsraum zu stark eingeschränkt ist

Nach Abschluss von Phase I sollte ein Pflichtenheft und ein Anforderungskatalog mit gewichteten Anforderungen erstellt sein.

4.4.2 Phase II - Erweiterte Papier-Evaluation

Marktübersicht verschaffen
In dieser Phase wird eine sogenannte *Long-List* erstellt. Es sollte sich eine Übersicht verschafft werden, welche Produkte überhaupt existieren.

Filterung der Long List
Nun wird die Long-List anhand der in Phase I erstellten Anforderungen gefiltert. Die daraus resultierende *Short-List* sollte ca. 2-3 Kandidaten umfassen.

Mithilfe der umgekehrten Pyramide können Lösungen sehr schnell aussortiert werden, da viele der Long-List die eine oder andere Grobanforderung nicht erfüllen und somit direkt aus der Liste fliegen.

Offerten einholen und beurteilen
Nun werden Offerten für die Lösungen der Short-List eingeholt. Diese Offerte kann verschiedenen Lösungen bieten:

- Alle Anforderungen erfüllt
- Lösung benötigt spezifische Erweiterungen und/oder Anpassungen
- Extremfall: Keine der Offerten sind passend und es muss eine Individuallösung her

Entscheid über weiteres Vorgehen
Nach dem Einholen und Prüfen der Offerten muss das weitere Vorgehen mit dem Management besprochen werden. Es sollte ein Empfehlung mit Vor- und Nachteilen (inkl. deren Lösung) präsentiert werden.

Es kann auch sein, dass keine der eingeholten Offerten den Anforderungen entspricht und man eine in-House Eigenentwicklung das Beste ist.
Das Ziel dieser Phase ist es, eine Lösung gefunden zu haben, die man nun Testen, Kaufen und Einführen kann. Es ist jedoch auch möglich, dass die Beste Lösung eine Eigenentwicklung ist. Falls das der Fall ist, fällt Phase III weg.

4.4.3 Phase III - Use Case Evaluation & Testing

Art der Hands-On Evaluation festlegen
In dieser Phase wird entschieden, welche Art der Evaluation durchgeführt wird:

- In-House, am eigenen System
- Extern, bei Experten

Vorteile	– Experten haben Erfahrung auf breiter Produktpalette – Experten haben bereits viele Evaluationen durchgeführt
Nachteile	– Kosten – Zeitaufwand – Offelegung von Firmendaten

- Third-Party / Referenzen

  Wie wird die Lösung bei anderen Firmen eingesetzt?

  Erfahrungen von Mitarbeiter anderer Firmen aufnehmen

- Produktpräsentation

  Keine "Hands-On" Evaluation

  Man muss den Aussagen der Vertreter Glauben schenken

Anwendungsfälle zusammenstellen
Bei welchen Vorgängen und Prozessen soll eine Hands-On Evaluation durchgeführt werden? Es sollte eine Liste mit allen Anwendungsfällen erstellt werden, bei welchen eine Hands-On Evaluation durchgeführt werden muss.

Solche Anwendungsfälle könnten Arbeitsschritte sein, die kritisch sind für das Unternehmen (z.B. Debitoren/Kreditoren Buchhaltung) oder auch solche, die im täglichen Alltag oft benutzt werden (Login)

Evaluation durchführen
Nun wird **für alle 2-3 Lösungen der Short-List** die Evaluation durchgeführt, indem die vorhin bestimmten Anwendungsfälle getestet werden.

Evaluationsbericht erstellen
Nun werden die Testergebnisse der Evaluation zusammengetragen, die Hauptkritikpunkte aufgestellt und zusammenfasst. Der Evaluationsbericht wird anschliessend in Form einer Kaufempfehlung dem Auftraggeber mitgeteilt.

Software kaufen und einführen
Die evaluierte Software wird nun gekauft, allenfalls noch angepasst und über eine bestimmte Periode im System eingeführt.

5 Platform Trends

5.1 Rekapitulation der Industrialisierung

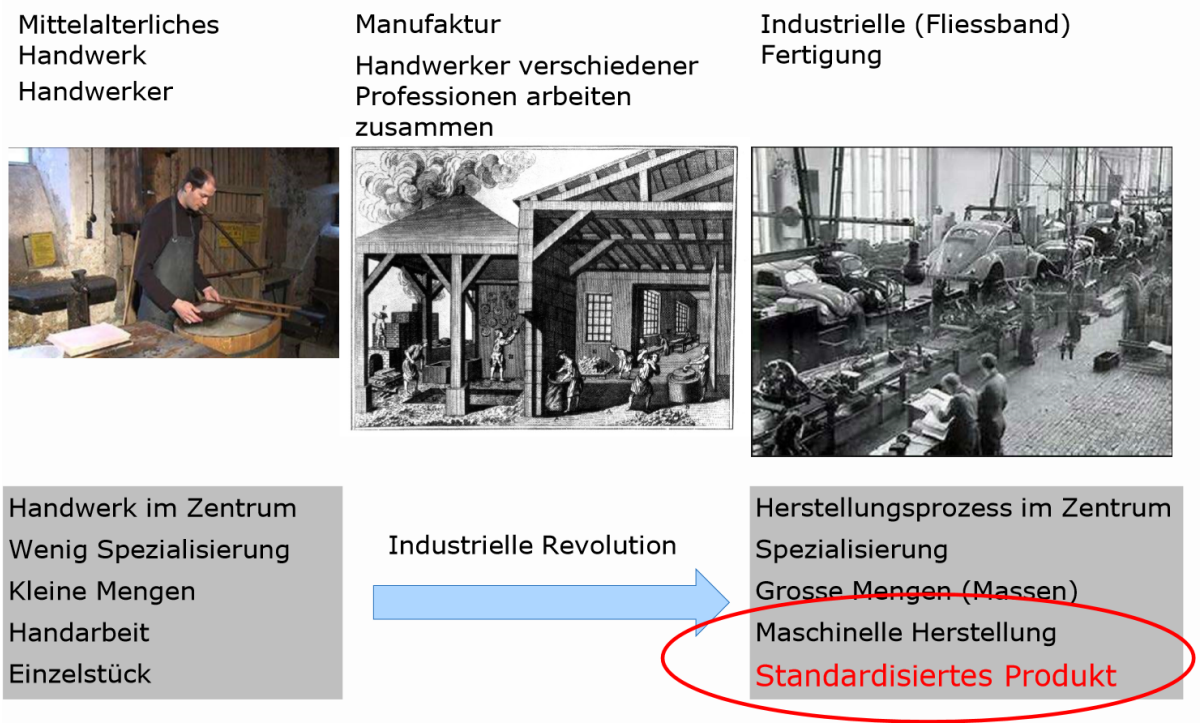


Abb. 5.1: Entwicklung des Handwerks vom Mittelalter bis heute

Auch die IT unterlief einer Industrialisierung, wenn auch wesentlich später als die 'normale' Industrialisierung. Die IT stützt sich heutzutage auch massiv auf Prozess- und Servicemodelle, Prozess Reifegrad- und Kontrollodelle ab, die grundsätzlich industrielles Gedankengut verkörpern.

5.2 Infrastrukturplattformen

5.2.1 Definition

Eine Infrastrukturplattform ist ein standardisierter und integrierter Satz von Infrastrukturkomponenten, Prozessen, Vorgaben und Richtlinien zum Betrieb einer bestimmten Klasse von Applikationen und umfasst:

- Hardware, OS, Middleware
- Einbindung in Netzwerk, Systemmanagement, Security-Infrastruktur und weiteren relevanten Umsystemen
- Vorgaben für Verteil-, Installations- und Betriebsprozesse
- Die dafür benötigten Werkzeuge und Prozesse

5.2.2 Aufbau

Plattformen bestehen normalerweise aus verschiedenen Schichten, wie in Abbildung 5.2 zu sehen ist. Die Layer sind so aufgeteilt.

Layer 0: Phyische und virtuelle Hardware / Hypervisor

Layer 1: Betriebssystem und Anbindung an die Umsysteme

Layer 2: Optionale Komponenten

Layer 3: Applications / Nutzlast

Layer 0-2 sind hier die tatsächliche Plattform und Layer 3 ist das, was auf der Plattform aufbaut, die Applikation und/oder Nutzlast

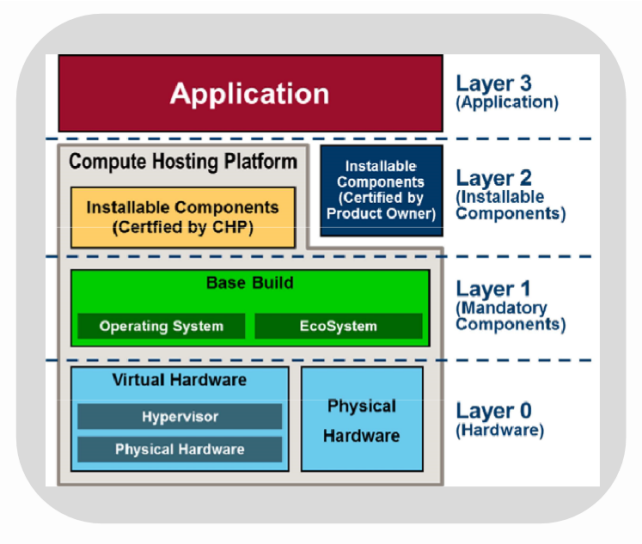


Abb. 5.2: Schichtenaufbau einer Infrastrukturplattform

6 Betriebliche Aspekte