Gute Überlegungen zu webshops gibt es hier:

<http://donauweb.at/online-marketing-blog/webshop-starten-mit-webshop-software/>

daraus einige wichtige Überlegungen:

Die wichtigsten Auswahlkriterien für ein **Online Shop System** sind:

1. Hat das **Online Shop System** erforderliche Schnittstellen zu **Warenwirtschaft** und/oder **CMS**?
2. Produktattribute: Wie einfach ist es zb ein T-Shirt in 5 Farben und 3 Größen im **Online Shop** einzugeben und zu verwalten? Das sind 15 Varianten. Können diese 15 Produkte eigene Produktnummern und Lagerstand haben? Wie leicht ist das zu verwalten? Wie wird die Auswahl für den Besucher dargestellt?
3. Welche **Funktionalitäten** benötigen Sie heute oder in bis zu drei Jahren? Gibt es **Plugins** für diese Funktionen?
4. Wie ist beim **Online Shop System** die Verfügbarkeit von Templates?
5. Gibt es eine **mobile Version** vom Shop?
6. Wie gut ist die **Internationalisierung**? (Mehrere Sprachen, mehrere Währungen, manche Produkte nicht in allen Ländern verfügbar)
7. Wie lange braucht ein Kunde **vom Warenkorb bis zu einer fertigen Bestellung**? Barrierefreiheit, Übersichtlichkeit und kurze Dauer sind wichtige Kriterien.
8. Wie hoch ist das Budget für die **Erstellung Ihres Online Shops**? Sowohl Entwicklungsbudget wie Hostingbudget steigen bei komplexen Shopsystemen.
9. Wie gut können Sie mit der Produktverwaltung und der Bestellverwaltung des **Online Shop Systems** arbeiten? Probieren Sie dies im **Online Shop** Ihrer Wahl aus, bevor Sie sich entscheiden.
10. Ist die Software bekannt und gibt es die Software in ein paar Jahren noch?
11. Gibt es eine Agentur mit Erfahrung mit Ihrem gewünschten Shopsystem in Ihrer Nähe?

Zalando baut sich zB seinen webshop selber (mit js, scala, php und hundert Entwicklern) und beschäftigt sogar Mathematiker die Vorschlagssystem für Kunden entwerfen, damit die beim nächsten Einkauf gleich Sachen präsentiert bekommen, die sie interessieren könnten (ist bei Mode nicht ganz einfach)

Welchen shop soll man nehmen:

Mietshops zB seoshop oder einen eigenen zB **WordPress** mit **Woocommerce**.

Auf Gesetz achten, AGBs müssen auch drinnen sein

# Wichtigste Prinzipien der Informationssicherheit (des Internetverkehrs)

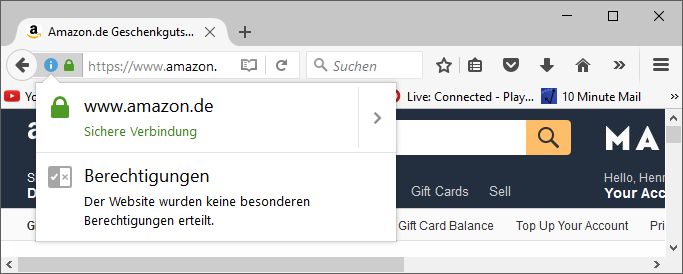
1. **Authentizität** (mit Zertifikaten erreicht)  
   Man kann sicher sein, dass das Gegenüber der ist, für den er sich ausgibt.
2. **Integrität** (private - public key Verfahren, Prüfsummen mitsenden)  
   Daten kommen unverfälscht an, wenn nicht, bemerkt man es.
3. **Vertraulichkeit** (Verschlüsseln)  
   Niemand kann den Datenverkehr mitlesen, der nicht dafür autorisiert ist
4. **Verfügbarkeit** (Redundante Systeme, Verfügbarkeitsrechnung, Load-Balancing)Verhinderung von Systemausfällen; der Zugriff auf Daten muss innerhalb eines vereinbarten Zeitrahmens gewährleistet sein.
5. **Verbindlichkeit** (Signieren, Protokollieren)  
   Ein System gewährleistet die Verbindlichkeit bzw. Zuordenbarkeit einer Menge von Aktionen, wenn es nicht möglich ist, dass jemand im Nachhinein die Durchführung einer solchen Aktion abstreiten kann. Verbindlichkeitseigenschaften sind besonders im Bereich des elektronischen Handels (engl. Electronic Commerce) und der elektronischen Geschäfte (engl. Electronic Business) von großer Bedeutung, um die Rechtsverbindlichkeit durchgeführter geschäftlicher Transaktionen (Kaufe, Vertrage etc.) zu garantieren.

## Zertifikat:

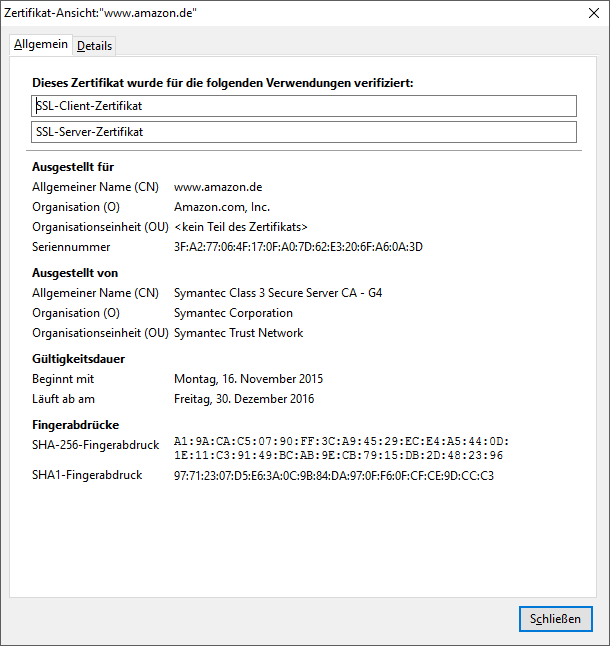
Durch ein Zertifikat lässt sich die Authentizität und die Integrität eines Servers überprüfen. Im Prinzip ist ein digitales Zertifikat nichts anderes als ein Datensatz, welcher mit einem privatem Schlüssel (**Private-Key**) signiert ist und zusätzlich bestimmte Eigenschaften wie *Aussteller* oder *Gültigkeitsdatum* vom Zertifikat enthält. Als weitverbreiteter Standard für das Erstellen von Zertifikaten gilt **X.509**. Mit einem **Public-Key** Verfahren überprüft man Gültigkeit des Zertifikates. Das heißt, man signiert das Zertifikat (des eigenen Webshops) mit dem eigenen Private-Key. Anschließend können Aufrufer der Webseite mit meinem Public-Key überprüfen, ob das Zertifikat tatsächlich gültig ist, da es das Gegenstück zum Private-Key sein muss.

Wichtig ist dabei, dass der public key auf einer vertrauenswürdigen Seite gelagert wird. Solche Stellen nennt man CAs, also Certification Authorities (zB lets encrypt, Symantec ist zwar gerade etwas in Veruf geraten, …)

Von jeder zertifizierten Website, welche mit **https** Daten übermittelt, kann man das Zertifikat im Browser einsehen:



Klickt man sich weiter kommt man zu folgender Übersicht vomZertifikat:



Erklärung:

* Allgemeiner Name (CN, Common Name) = Name des Webseitenbesitzers (ausgestellt für) bzw. Name der CA (ausgestellt von)
* Organisation = Name des CA Unternehmens bzw des Besitzers der Website
* Organisationseinheit (OU, Organisation Unit) = Abteilung der Unternehmens
* Seriennummer = Wird von der CA vergeben. Laut x.509 Standard sollte der Name der CA in Kombination mit der Seriennummer eindeutig sein. Praktisch ist dies nicht so wichtig und wird oft von den CAs vernachlässigt. (Quelle: [Wikipedia](https://de.wikipedia.org/wiki/X.509))
* Fingerabdruck = Der Hash-Wert, der sich aus dem ganzen Zertifikat (also den Werten von CN, O, OU, Seriennummer, Gültigkeitsdauer, …) ergibt. Dieser Fingerabdruck ist weltweit eindeutig. Er wird dazu verwendet, um die Zertifikate zu identifizieren. (Das geht ja dann schnell, weil hashing). Ein Aussteller kann zusätzlich noch den Hashing Algorithmus wählen (SHA-1, SHA-256, MD5,…). (Quelle: [Link](http://morgansimonsenblog.azurewebsites.net/2013/04/16/understanding-x-509-digital-certificate-thumbprints/))

Zertifizierungsstellen (Certificate Authority, CA) stellen Zertifikate aus. Es gibt circa 700 Stellen. Ein Zertifikat kann verschiedene „Levels“ haben, die beschreiben, wie sehr man der zertifizierten Person bzw. Organisation vertrauen kann. „Level 1“ bedeutet, dass die CA überprüft, ob man wirklich die Person ist, zu der die Domain gehört, wofür man das Zertifikat braucht. Je höher das Level, desto genauere Überprüfungen finden statt. zB bei letsencrypt für die Schule wurde nur überprüft ob der Server klio wirklich zur Domäne htl-villach.at gehört und die Domäne zu den IP Adressen passt, die beim Provider eingetragen sind.

CAs sind im Browser eingetragen (mit Namen und Public-Key). Sobald ein Zertifikat einmal überprüft wurde, gilt es im Browser als gültig. Man braucht also keine weiteren Überprüfungen.

Zusätzlich gibt es auch verschiedene Möglichkeiten, wie man seine Website zertifizieren lassen kann:

* Simple Site Zertifikate e
  + Für eine fixe Domain wie „www.example.com“
  + Billigste Variante.
* Stern-Zertifikate
  + Z.B. „\*.example.com“
  + Also ist es möglich, dasselbe Zertifikat für „an.example.com“, für „hello.example.com“, für „asdf.example.com“ zu verwenden.
  + Dafür ist so ein Zertifikat aber auch teurer, als Simple Site Certificates.
* Selbst signierte Zertifikate
  + Da man sie selber macht, kosten sie auch nichts.
  + Quasi wie ein falscher Führerschein. Man benutzt selbst signierte Zertifikate für Tests.
  + Kein Browser erkennt diese Zertifikate an, außer man fügt diese händisch in die Zertifikat-Tabelle des Browsers hinzu.

## HTTPS/SSL/TLS

HTTPS („S“ steht für „Secure“) ist kein eigenes Protokoll.

Daten werden verschlüsselt aber trotzdem mit HTTP übertragen. Normalerweise wird dabei der Port 443 verwendet. Verschlüsselt werden die Daten mit TLS (Transport Layer Security). Heutzutage sagt man zu TLS oft noch SSL, wobei TLS die Erweiterung zu SSL ist.

Warum benötigt man ein Zertifikat, wenn man einen sicheren Datenaustausch mit HTTPS gewährleisten möchte?

Eine Verschlüsselung besteht aus der Verschlüsselung der Daten beim Sender (Client) und der Entschlüsselung der Daten beim Empfänger (Server). Bei SSL bzw. TLS arbeitet man mit zwei unterschiedlichen Schlüsseln zur Ver- und Entschlüsselung. Das sogenannte Schlüsselpaar besteht aus einem privaten (Private Key) und einem öffentlichen Schlüssel (Public Key). Der öffentliche Schlüssel des Empfängers ist dem Sender bekannt. Er benutzt ihn zum Verschlüsseln der Daten. Anschließend können die verschlüsselten Daten aber nicht mehr mit dem öffentlichen Schlüssel entschlüsselt werden. Dafür braucht es den privaten Schlüssel, der nur dem Empfänger bekannt sein darf und deshalb zwingend geheim gehalten werden muss. Nur der Server mit dem passenden privaten Schlüssel ist in der Lage die verschlüsselten Daten zu entschlüsseln (asymmetrisches Verschlüsselungsverfahren bzw. Public-Key-Verfahren).

Bevor nun der Sender Daten verschlüsseln darf, muss er jedoch zweifelsfrei feststellen, ob der öffentliche Schlüssel, den er vor der Verschlüsselung vom Empfänger mitgeteilt bekommt, auch tatsächlich dem Empfänger gehört, dem er die Daten verschlüsselt schicken will. Eine per SSL verschlüsselte Verbindung bietet keinen Schutz, wenn nicht **sichergestellt ist, dass der öffentliche Schlüssel von dem Server kommt, zu dem eine verschlüsselte Verbindung hergestellt werden soll**.  
An der Stelle kommt jetzt das Zertifikat ins Spiel, **mit dem sich ein Server und sein öffentlicher Schlüssel authentisieren**. Um die Gültigkeit des öffentlichen Schlüssels zu unterstreichen, lässt sich der Server-Betreiber und Domain-Inhaber ein Zertifikat ausstellen, in dem unter anderem der Domainname, der öffentliche Schlüssel, ein Ablaufdatum und welche Instanz die Vertrauenswürdigkeit bestätigt hat enthalten sind. Durch das Zertifikat authentisiert sich der Empfänger gegenüber dem Sender, bzw. der Server gegenüber dem Client. Gleichzeitig kann der Client das Zertifikat überprüfen (Validierung) und somit die Vertrauenswürdigkeit feststellen (Authentizität). (Quelle: [elekronik-kompendium](http://www.elektronik-kompendium.de/sites/net/0902281.htm))

## Let’s Encrypt

Let’s Encrypt ist ein Tool, welches den Zertifizierungsvorgang automatisiert.

Vorgang:

1. *Man downloadet sich den Let’s Encrypt Agent auf den Webserver, der die zu zertifizierende Website hostet:  
   sudo apt-get certbot-auto*
2. *Man gibt folgenden Befehl ein:  
   ./certbot-auto -d example*.com
3. …
4. Profit!  
   (Quelle: [certbot&letsencrypt doku](https://certbot.eff.org/docs/intro.html))

Man muss quasi nichts über Zertifikate wissen.

Was im Hintergrund passiert:

1. Der Agent generiert ein Schlüsselpaar (Private-Public Key) und sendet den Public Key an eine (bestimmte) CA
2. Die CA registriert den Key
3. Der Agent möchte die Domain „example.com“ verifizieren.
4. Die CA fordert vom Agent eine oder mehrere „Challenges“  
   Zum Beispiel:
   1. Die CA schickt dem Agent ein Textfile und eine „nonce“.

(Der Inhalt des Textfiles und die Nonce ist bei der CA unverschlüsselt hinterlegt.)

* 1. Der Agent stellt das File, mit dem (vorher generieten) Private Key verschlüsselt in ein bestimmtes Verzeichnis am Webserver (sodass man es mit „example.com/irgendwas“ aufrufen und downloaden kann)
  2. Die nonce muss der Agent auch noch mit dem Private-Key verschlüsseln.

1. Nachdem der Agent die Aufgaben erledigt hat, sagt er der CA, dass er fertig ist.
2. Die CA überprüft, ob die Aufgaben erfüllt worden sind  
   Beispiel von oben:
   1. CA downloaded das mit dem Private-Key verschlüsselte Textfile und entschlüsselt es. Wenn das drin steht, was die CA ursprünglich versendet hatte, dass ist diese Aufgabe erledigt.
   2. Die CA entschlüsselt die mit dem Private-Key verschlüsselte nonce und überprüft sie mit der ursprünglich hinterlegten, unverschlüsselten nonce auf Gleichheit.
3. Ist alles gut gelaufen, generiert der Agent ein weiteres, Long-Term Private-Public-Key Paar und sendet der CA wieder den Public Key.
4. Die CA vergibt nun ein x509 Zertifikat
5. Der Agent installiert das Zertifikat und setzt ggf. nötige Optionen am Webserver, damit https verwendet wird

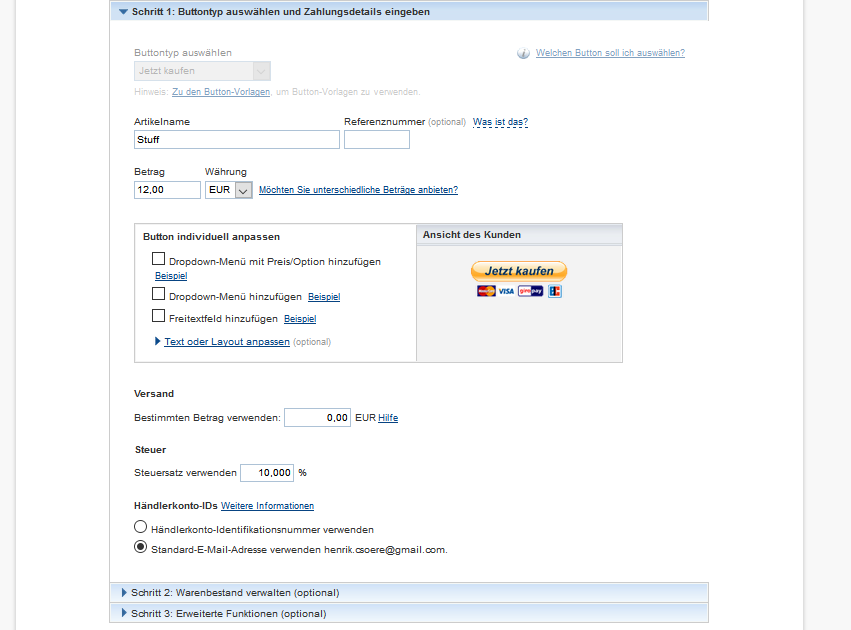
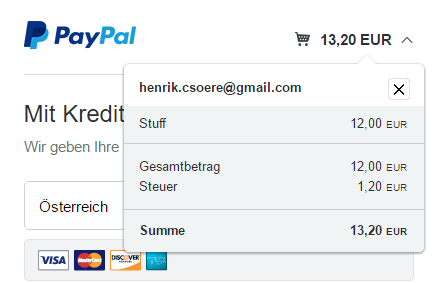
(Quelle: [cryptology.net](https://www.cryptologie.net/article/274/lets-encrypt-overview/))

## Bezahlungsmethode PayPal am eigenen Webshop einrichten:

PayPal (ursprünglich von Elon Musk, dem Tesla Chef erfunden) ist eine der gängigsten Methoden, um Online zu Shoppen. Um PayPal als Zahlungsmethode für den eigenen Webshop einzurichten kann man sich jederzeit Bezahl-Buttons generieren lassen, und den HTML-Code dafür auf seinen Webshop hinzufügen.

Man geht dazu einfach in die Übersicht von seinem PayPal Account auf „Verkäufereinstellungen“ und wählt dort den Punkt „Meine Zahlungsbuttons verwalten“ aus. Dort kann man verschiedene Arten von Buttons aussuchen, die Preise einstellen, usw.

Man muss also lediglich den HTML-Code des Buttons auf die gewünschte Position des Webshops hinzufügen.



Oder man nimmt Anbieter wie wirecard, die gleich plugins für mehrere Zahlungsmethoden (wie Kreditkarten, eps Überweisung und so anbieten). Dafür zahlt man aber Gebühren, die üblicherweise vom Umsatz abhängig sind.