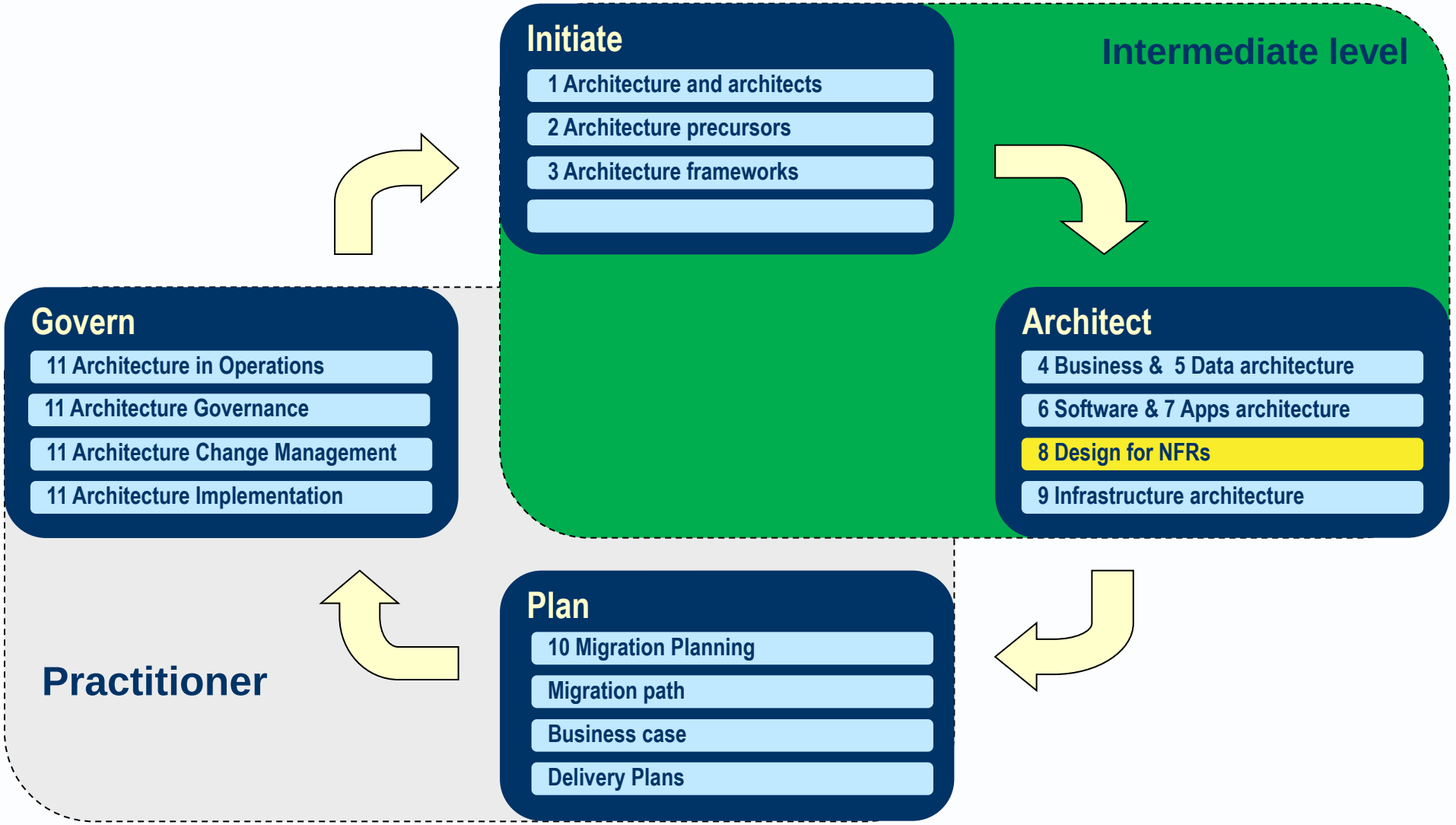
Avancier Reference Model

Design for Qualities / NFRs (ESA 8)

8. Design for NFRs

Avancier



8: Design for Qualities (NFRs)

- This section contains a selection of measures and common techniques that could be relevant in answers to exam questions.

8.1: Non-functional requirement types (NFRs)

- **PARRISSS**
- **UMPIE**

Performance

- [a requirement type] that is subdivided into two measures.

- Throughput: volume or number of services performed in a time period.
- Response or cycle time (aka latency): time taken from request to response or completion.

- Sometimes, improving one measure damages the other.

PARRISSS

- **Availability** [a requirement type] the amount or percentage of time that the services of a system are ready for use, excluding planned and allowed down time.
- Possible measures include MTBF / (MTBF + MTBR), which usually refers to availability at the primary site, excluding disasters.

- **Reliability** [a requirement type] the ability of a discrete component or service to run without failing.
- Possible measures include mean time between failures (MTBF).
- Aside: The measures above are sometimes applied only to platform applications, ignoring faults and failures in business applications.

- **Recoverability** [a requirement type] the ability of a system to be restored to live operations after a failure.
- Possible measures include mean time to repair (MTBR). Usually refers to disaster recovery using resources at a remote site.

PARRISSS

- **Integrity** [a requirement type] a term with four possible meanings defined under “Data Integrity”.
- **Scalability** [a requirement type] the ability for a system to grow with increased workloads.
- **Security** [a requirement type] the ability to prevent unauthorised access to a system.
- **Serviceability** [a requirement type] the ability to monitor and manage a system in operation.

UMPIE

- **Usability** [a requirement type] the ability of actors to use a system with minimal effort.

- Aside: Measures include PLUME.
  - Productivity; tasks completed in a given time.
  - Learnability; how much training to reach a proficiency level.
  - User Satisfaction; scores given by users.
  - Memorability; how long to forget how to use the system.
  - Error Rates.

UMPIE

- **Maintainability** [a requirement type] the ability to analyse, then correct or enhance a system.
- **Portability** [a requirement type] the ability to move a system from one platform to another, or convert it to run on another platform.
- **Interoperability** [a requirement type] the ability for systems to exchange data using shared protocols and networks.
  - It may embrace also “integratability” - the ability of interoperable systems to understand each other, which requires either common data types or translation between data types.
- **Extensibility** [a requirement type] the ability to add new features; a kind of maintainability.

8.2: Design for NFRs (bar security)

Design for response time

- [a technique] that may involve minimising
  - distribution,
  - network hops,
  - database accesses, and
  - message queue length.

- The general advice is to look for bottlenecks and tackle them one by one.

Design for throughput

- [a technique] that usually involves running processes in parallel.
- **Scale up** [a technique] that increases the power of one node. Usually means adding resources, processors or memory. Generally good for speed and throughput.
- **Scale out (aka clustering)** [a technique] that increases the number of parallel nodes, sometimes adding more nodes to a cluster. Usually requires some kind of load balancer to sit in front of the cluster and distribute service requests between them. Generally good for throughput. Not always good for response time.
- **Caching** [a technique] that places copies of persistent data in a location nearer to the user than the original data source. Generally good for response or cycle time. Can raise concerns about data integrity and security.

Database optimisation

- [a technique] for reducing needless database access that include
  - Normalisation
  - Denormalisation
  - Indexes
  - access path analysis.

Database optimisation

Avancier

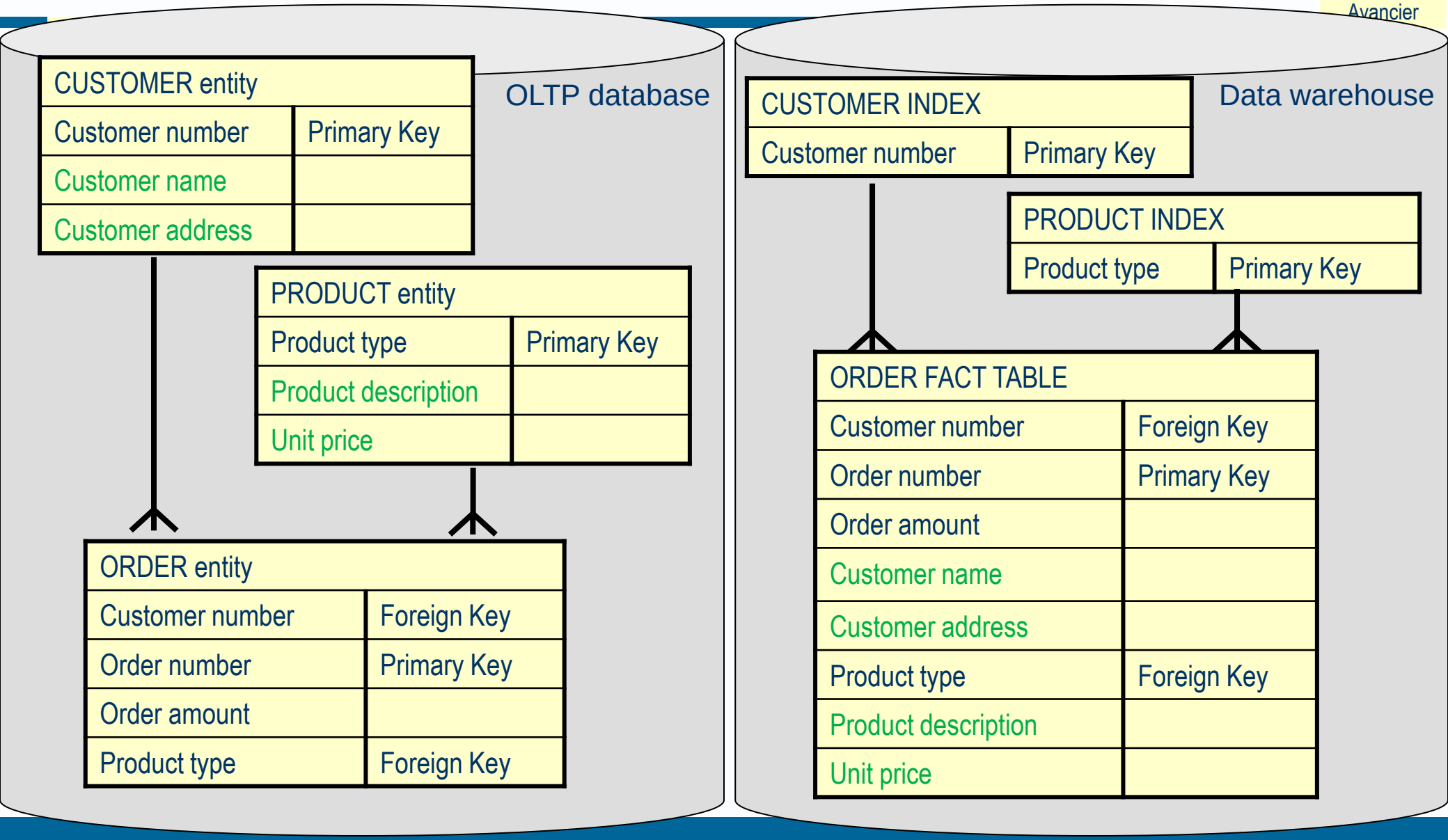
Normalisation

- Relational data analysis.
- A technique for defining a data store structure that assists data integrity by storing each fact once.
- It also optimizes update processes by minimising redundant data storage.

Denormalisation

- A design technique that optimizes input and/or output processes by structuring a data store structure to reflect the most important input or output data flow structures, at the expense of duplicating some stored data.

Separation of update and reporting databases



Database optimisation

- **Indexing** [a technique] that creates a list of pointers to stored data elements.
- Indexes can help to optimise batch input and output processes, which typically run overnight.
- Indexes may be temporarily disabled during the day to optimise on-line update processes.

- **Access path analysis** [a technique] that studies the route a process takes through a data store structure.
- A very common source of performance problems is that an SQL programmer does not know the access path their procedure takes through a database.
- So it is advisable to use access path analysis and/or employ highly skilled SQL resources for critical database access programs.

Design for resilience (availability and reliability)

- [a technique] that builds redundancy into the system, with parallel active components or fail over to a passive standby component. E.g. to scale out, or add one to the number of servers in cluster that calculation or prototyping suggests is needed. Other techniques include defensive design

- **Fail over** [a process] that automatically switches over to a redundant or standby system, upon the failure or abnormal termination of the previously active system. Failover happens suddenly and generally without warning.

Defensive design

- [a technique] that means either
- 1. Designing a client component so that it does not fall over if a server component does not work properly; asynchronous invocation may help.
- Or
- 2. Designing a server component so it does not depend on input data being valid, which means testing input data and preconditions before processing (and is the opposite of “Design by Contract” as promoted by Bertrand Meyer.)

Design for recoverability

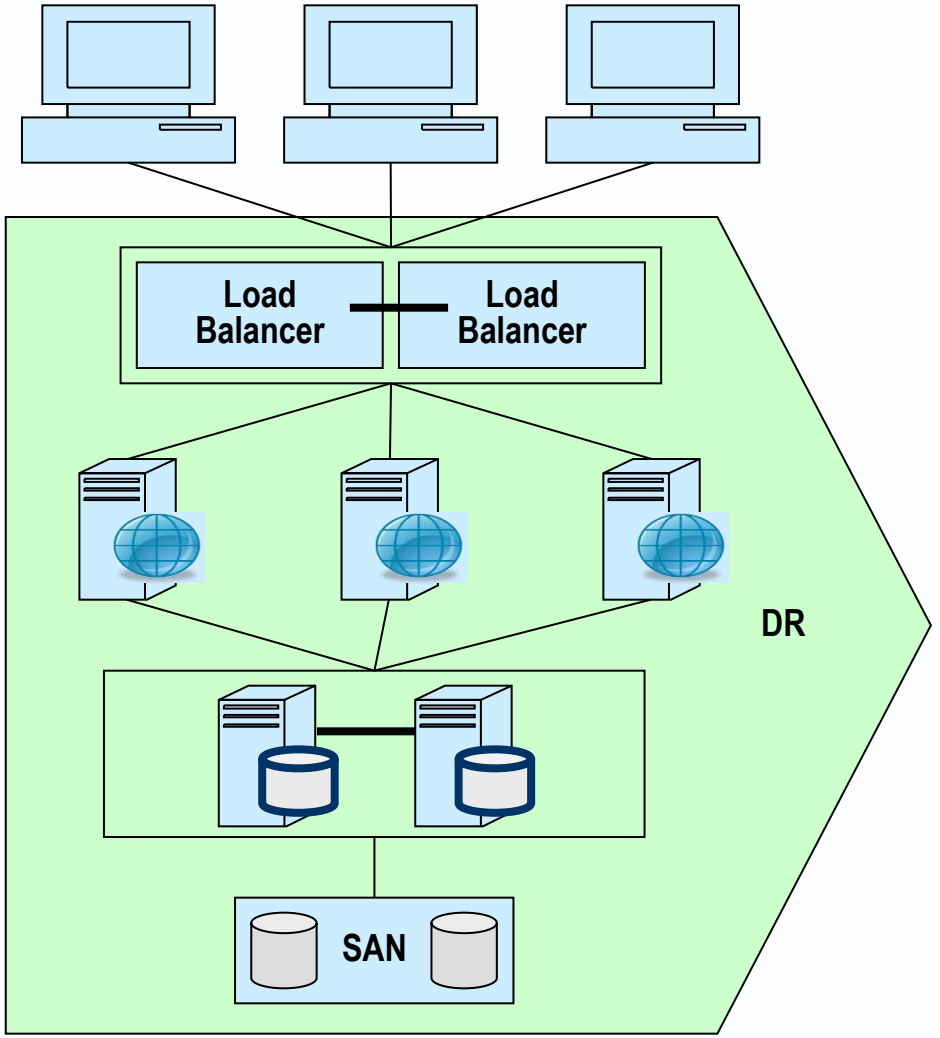
- [a technique] to back up and provide some kind of switch-over or fail-over procedure. Procedures must address also “fail back”, to return operations from a disaster recovery site to the normal production site.

- **Backing up** [a process] to make a copy of data, so that it can be used to restore the original after data loss. Used in disaster recovery. Also used to restore individual files that have been deleted or corrupted. Backups are typically the last line of defence, coarse-grained and can be inconvenient to use.

- **Backup site** [a location] where systems are or can be duplicated. A cold site has no equipment. A warm site has infrastructure but no up-to-date data or software. A hot site has up-to-date software and more or less up to date copies of data.

Design for recoverability

- **Synchronous replication**
  - replicates data storage in real time
    - needs high bandwidth and low latency
    - limited by distance 150K?.
- **Asynchronous replication**
  - replicates data storage off-line
    - makes and keeps copies of data at a remote site
    - operations can be resumed at the remote location using a remote copy.
    - not subject to distance limits.



Design for integrity

- [a technique] such as
  - reducing data replication,
  - normalising stored data,
  - switching on automated referential integrity checks,
  - applying transaction management to update processes,
  - removing caches, and
  - consolidating distributed databases.

Design for serviceability

- [a technique] to instrument applications so that they report on what they are doing, and how well they are doing it.

8.3: Design for Security

- **Security architecture** [a view] showing design features designed to protect a system from unauthorised access - to maintain the required data qualities of confidentiality, availability, and integrity.
- Not a cohesive architecture domain on its own so much as features added to other views.

Design for human and organisational security

- [a technique] defining anything that can be done outside of IT systems to secure business information, such as
- security guards,
- locks on doors,
- definition and roll out of policies and procedures.

Data security

- [a concern]

- 1: Confidentiality alone.
- 2: A combination of Confidentiality, Integrity and Availability.

- Tom Peltier suggests rating the security level of a data item, data structure or data store as equal to the highest of the individual ratings (high, medium, low) awarded for Confidentiality, Integrity and Availability.

- **Information domain** [an entity] a uniquely identified set of items with a security policy that defines the rules that constrain access to data within the domain.

- **Security policy** [a document] that defines which actors have access rights to which data objects in a given information domain – and which security processes or properties are used.

- **Identity** [a property] one or more data items (or attributes) that uniquely label an actor. E.g. passport number or user name.

- **Encryption** [a process] to encode data items (in a data store or data flow) so that they are meaningless to any actor who cannot decode them.

- **Checksum** [a property] a redundant data item added to a message, the result of adding up the bits or bytes in the message and applying a formula. This enables the receiver to detect if the message has been changed. It protects against accidental data corruption, but does not guarantee data flow integrity, since it relies on the formula being known only to sender and receiver.

- **Digital signature** [a property] a cryptographic device that simulates the security properties of a handwritten signature. More secure than a check sum, it is said to guarantee the data flow integrity of a message, since the signature is corrupted if the message content is changed.

Design for applications security

- [a technique] for preventing unauthorised use of an application.
- **Identification** [a process] via which an actor supplies their identity to an authority. It is usually followed by authentication.
- **Authentication** [a process] to confirm or deny that an actor is trusted - is the entity to which an identity was given.
  - E.g. A password check. It produces one of four results: true positive, true negative, false negative - which leads to wrongly-denied access, or false positive - which leads to unauthorised access. It is usually followed by authorisation.
- **Authorisation** [a process] for giving access to a trusted actor, based on that actor's known access rights. It is usually followed by access.
- **Access** [a process] that locates data or processes of interest and retrieves them for use.

Three-factor authentication

- **Three-factor authentication** [an authentication] that checks what users remember (e.g. password, mother's maiden name) and carry (e.g. credit card or key) and are (using biometric data).

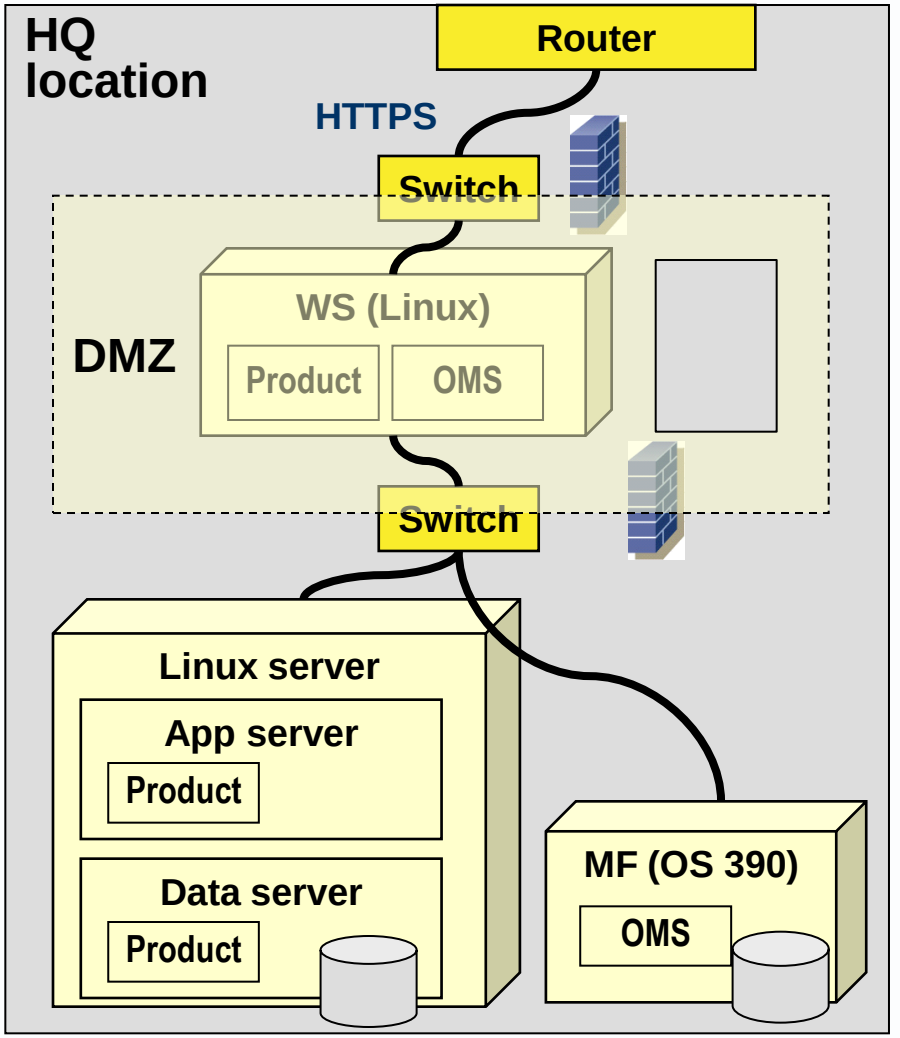
- 3 factor NOT = identity + password + key

Design for infrastructure security

- [a technique] for protecting client and server computers from malicious access.

- **Firewall** [a component] at the boundary of a network that can detect, filter out and report messages that are unauthorised and/or not from a trusted source.

- **DMZ: De-Militarised Zone** [a component] of a network, usually between the public internet and the enterprise network.
- It uses firewalls to filters out messages that fail security checks.
- It contains servers that respond to internet protocols like HTTP and FTP.



HTTPS

- [a process]
- a normal HTTP interoperation over
  - an encrypted Secure Sockets Layer (SSL) or
  - Transport Layer Security (TLS) connection.

- This ensures reasonable protection of data content from those who intercept the data flow in transit.

- (Aside: If an HTTPS URL does not specify a TCP port, the connection uses port 443.)

Web site security

- [a process] whereby, for example, a web browser checks the public key certificate of a web server at the other end of an HTTPS connection.

- The aims are to check the web server is authentic (who it claims to be) and that messages to/from with the web server cannot be read by eavesdroppers.

LEFT OVERS

Design for NFRs - Security

Avancier

Design for security	Design for security is addressed elsewhere in this reference model under the headings of business, data, applications and infrastructure architecture..
ISO/IEC 17799	Information technology: Code of practice for information security management.
ISO/IEC 24762:2008	Information technology — Security techniques — Guidelines for information.
ISO/IEC 27001	Information technology — Security techniques — Information security management systems — Requirements.

Security: Public key certificate

Avancier

Public key certificate

An electronic document that enables a web server to accept https connections, or [a client?] to verify that a public key belongs to an individual.

It incorporates a digital signature to bind together
a public key with
an identity (the name of a person or an organization, their address, and so forth).

Encrypting File System	Encrypting and decrypting documents.
File recovery	Recovering encrypted files if the EFS certificate is accidentally deleted or damaged.
Server authentication	Verifying the identity of a server to computers that are connecting to it.
Client authentication	Verifying the identity of a computer to a server it is connecting to.
Secure e-mail	Encrypting and digitally signing e-mail.
Code signing	Verifying the publisher of a program. For example, if you download an ActiveX program, its digital signature verifies that it is published by the organization that is listed as the publisher.

Security: Certificate authority

Avancier

Certificate authority	The web site administrator must get a public key certificate signed by a certificate authority. This signature certifies (authenticates) that the certificate holder is the entity it claims to be. Web browsers are generally distributed with the signing certificates of major certificate authorities, so that they can verify web-server certificates.