Introduction

Architecture

Key points:

- There are multiple architectures in a system.
- Architecture should be expected to change over time

Architecture is:

- A subjective shared understanding of a system's design by the expert developers on a project.
- Defined by the form of major components of the system and how they react.
- "The important stuff - whatever that is"

Enterprise Architecture

Complexity of the architecture typically mimics the complexity of the business logic.

Application Performance

Performance needs to be considered on an application basis.

Try to not over optimize to the point the application becomes difficult to work with.

Most common things to consider regarding performance:

- **Response time** - how long the system takes to process a request.
- **Responsiveness** - how quickly the system acknowledges a request.
- **Latency** - the minimum time required to get any type response even if the work to be done does not exist.
- **Throughput** - how much work can be done in a given amount of time. Performance is either throughput OR response time whichever is more important in the context of the application. From a users perspective responsiveness may be more important than response time. In this case improving responsiveness at the cost of response time will increase performance.
- **Load** - is a statement of how much stress a system is under which can be measured in several ways. Load is usually a context for some other measurement like response time. One way of measuring load is by the number of concurrent users. You might say that response time is .5s for 10 users and 2s for 20 users. Load being represented as users providing context for response time.
- **Load Sensitivity** - is an expression of how the response time varies under a given load.
  - Say system A has a response time of 0.5s for 10 through 20 users and system B has a response time of 0.2s for 10 users that rises to 2s for 20 users.
  - In that case system A has lower *load sensitivity* than B.
  - This can be referred to as system degradation.
  - System B degrades more than system A.

- **Efficiency** - is performance divided by resources.
    - A system that gets 30 tps on two CPUs is more efficient than a system that gets 40 tps on four CPUs.
- **Capacity** - Maximum effective throughput or load. This might be an absolute maximum point of performance or a point where performance drops below an acceptable level.
- **Scalability** - a measure of how adding resources (typically hardware) affects the performance.
    - Vertical scalability means adding more power to a single server
    - Horizontal scalability means adding more servers
    - When building enterprise systems it's usually better to prefer scalability rather than capacity or efficiency.
    - Scalability gives you the option to scale on demand when needed.

Patterns