Microservices

Divide, until each part of the problem are:
- solvable separately
- manageably small

The upgrade/replacement of one service should not require co-ordination with another.

MODULE: bounded, contigous group of statements having a single name by which it can be refered to as a unit

COMPONENT: things that are independently replaceable and upgradeable

- organize around business capabilities
- products not projects
- decentralized governance
- decentralized data management
- design for failure
- smart endpoints and dumb pipes
- monitorization and automation

(A successful business is not static)

The domain will change as the business change.

constant gardening

SERVICE: component that exists in its own process,
clients talk to it over some inter-process communication mechanism: RPC, REST/over HTTP, messaging, etc

service
eah layer providind a narrow and well-defined
interfaces to the layers around it
services / clients

Keep what changes together, together

allow ACID requirements to be satisfied inside, within ~~the~~ one boundary of one service only