

So you want to build an enterprise SaaS product?

Mark Nelson
EVP Product Development



SaaS is all about the second S



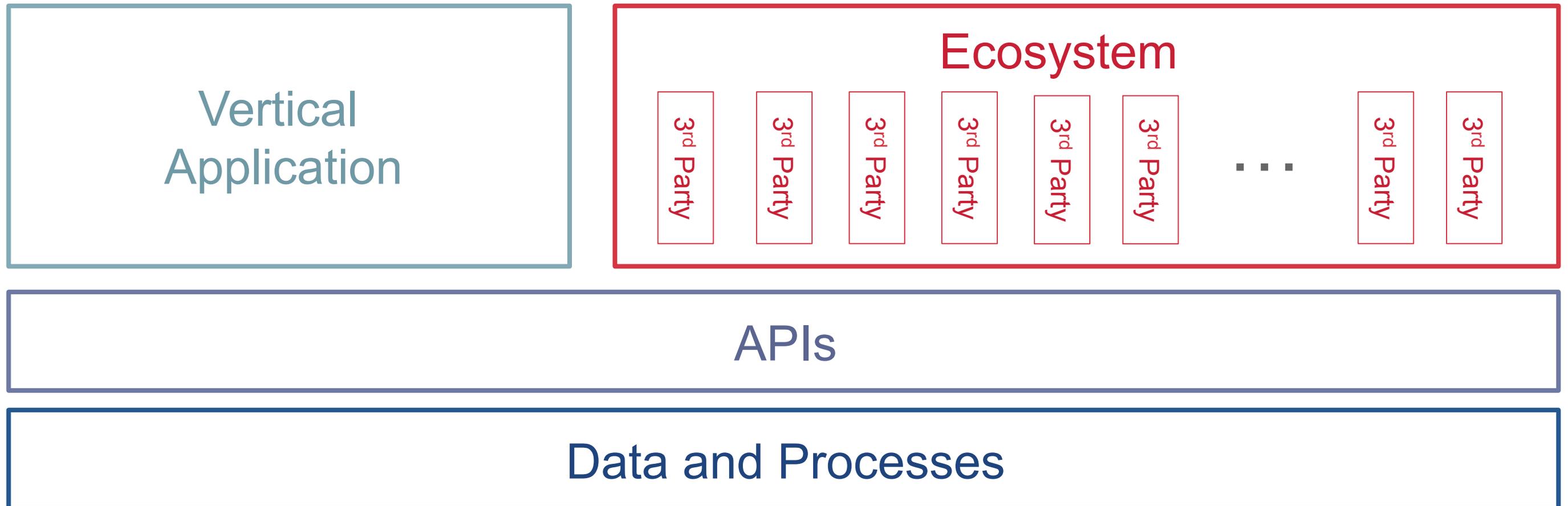
... is indistinguishable from...



Make some business activity easy



Build a system, not just an application



Consumer Grade User Experience



SaaS Service Expectations

Always on reliability



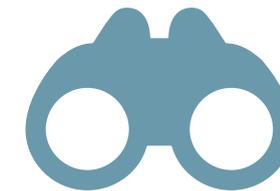
Security & privacy



Performance



Visibility & transparency



Foundational Principles

- Scale and speed requires decentralization and empowerment
 - Decompose system into a set of services
 - Dedicated team owns each service
 - Team is enabled and responsible for every aspect of that service
- Simplicity, simplicity, simplicity



Failure is Not an Option

- Customers rely on services being up – always
- Every component fails. Humans make mistakes; under stress make more mistakes
- Design system with reality in mind and test failure and recovery before they are real
- Learn from every event. Ask “why?” until you know the foundational problems



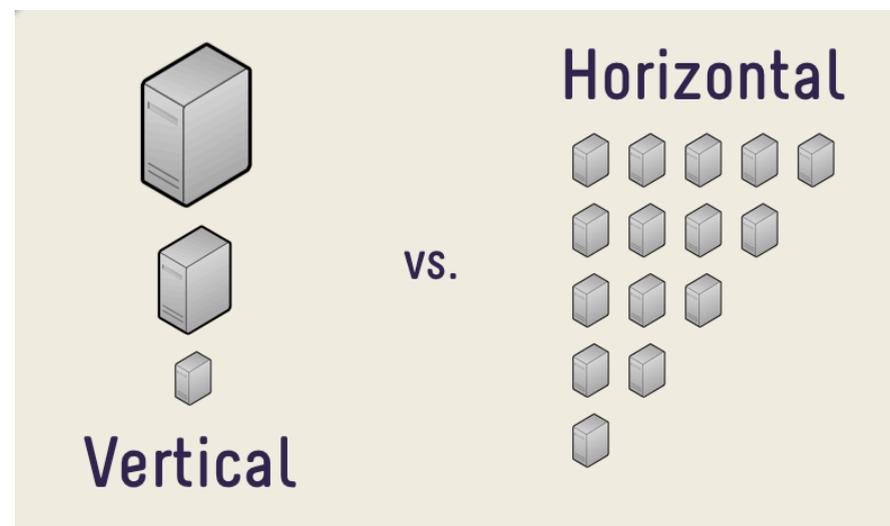
Horizontal vs. Vertical Scaling

Horizontal Scaling

- Small building blocks is cheaper & easier
- Adds redundancy to your system
- Requires that you design the E2E system – request routing, failover, provisioning, upgrade

Vertical Scaling

- Easy to use
- Requires ever increasing levels of “magic” technology
- Concentrates the failure points



Failure is Not an Option

- State comes in many forms
- State needs to be recoverable or re-createable
- Store recoverable state in systems are designed to store state
- Automate the recreation of all other state
- EC2 forces this model because local storage is ephemeral



Automation – Pets Versus Cattle

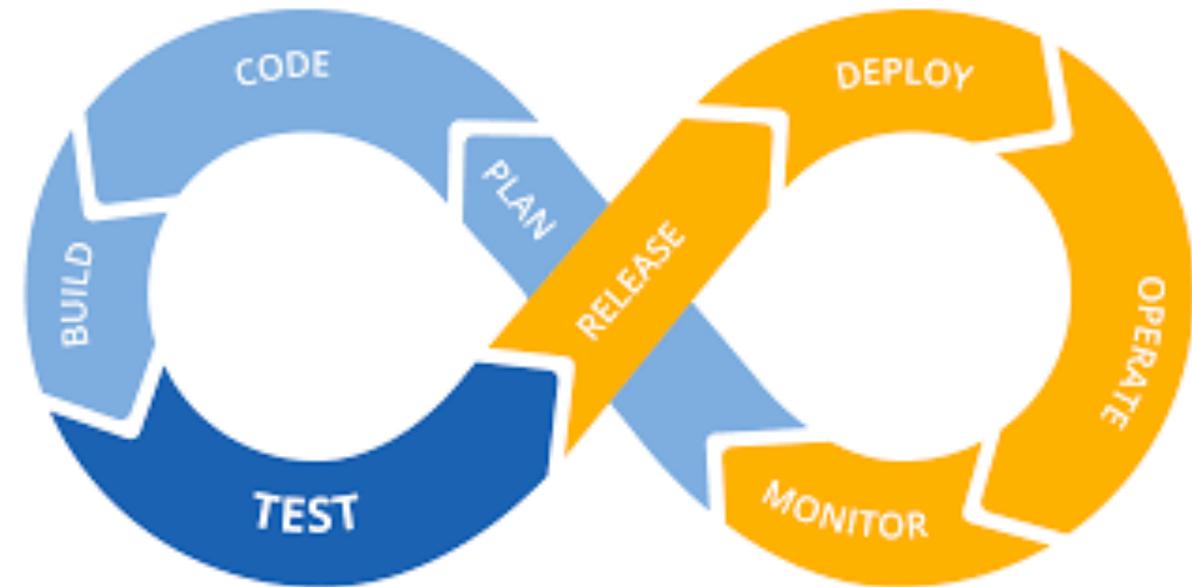


- Pet machines are precious entities
- You give them names such as pussnboots.concurtech.org
- You raise them lovingly
- When they get sick, you nurse them back to health

- Cattle machines are commodities
- You give them numbers such as sl4379.concurtech.org
- You mass-produce them with automation
- When they get sick, you shoot them and produce another

E2E Team Ownership

- Enable teams with tools
- Give them an envelope (set of rails) they have to live within
- Make them accountable and responsible for the rest
- Planning, architecture, deployment, monitoring, operations, security...



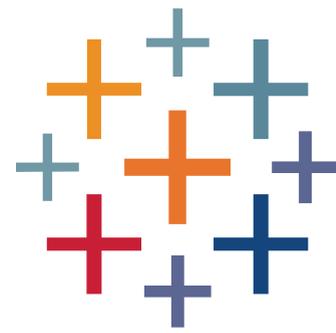


DOING AGILE APPEARS EASY

BEING AGILE TAKES DISCIPLINE AND COURAGE

Software is the Easy Part: Engineering Culture at Scale

- How to enable freedom without resulting in chaos?
- What is the right level of command & control that allows teams to be empowered and responsible for their choices but maintains necessary consistency?
- Communication, communication, communication
- Technology is the result of the culture that builds it
- An engineering culture must be able to continuously learn and change; any decision you make today is already wrong or will be soon
- Learn from others:
 - [Etsy](#), [Amazon](#), [Netflix](#), [RSA](#)



+ a b l e a u[®]