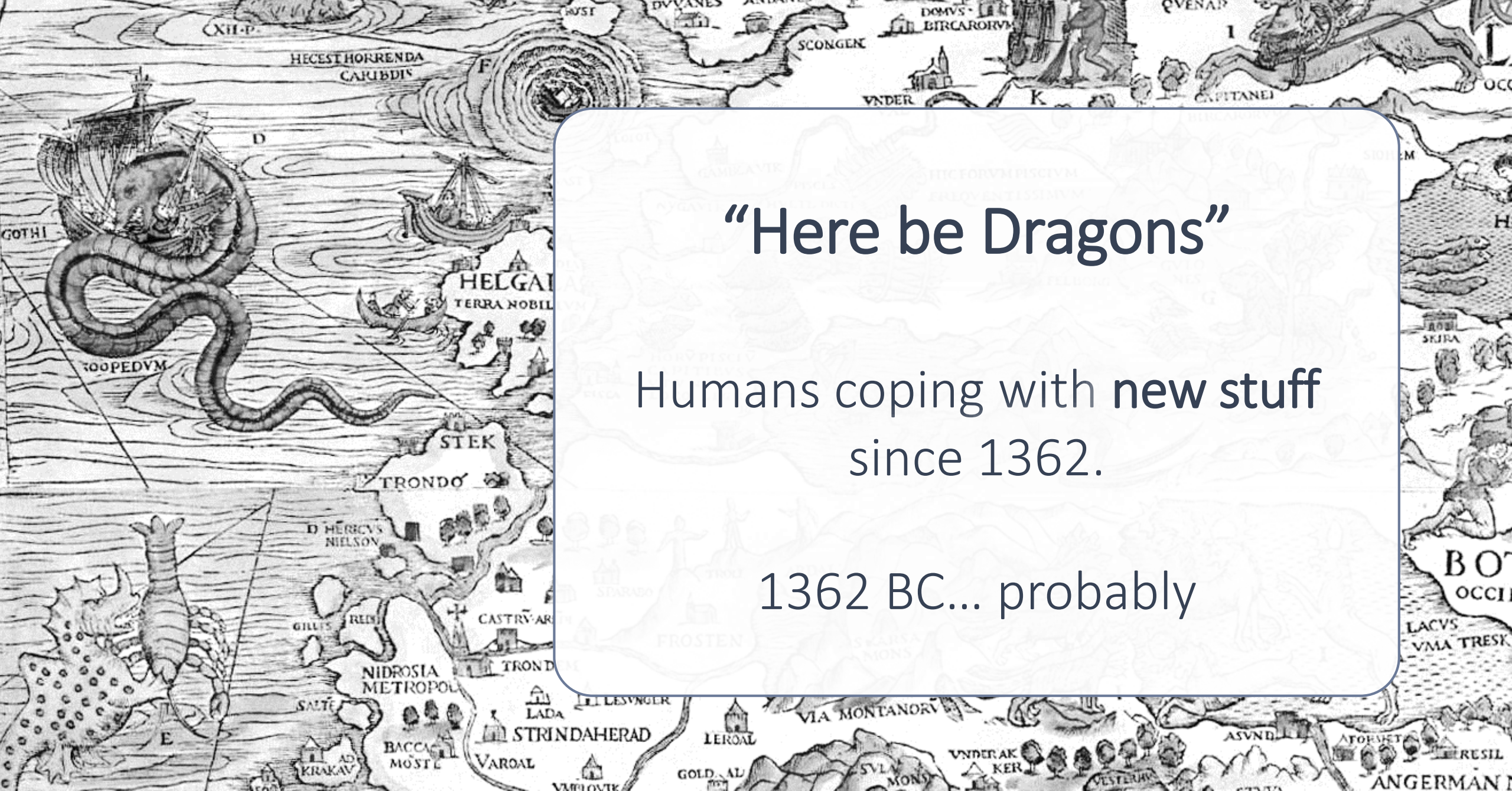


Learning Stuff the Hard Way

Suren Pillay

February 2017



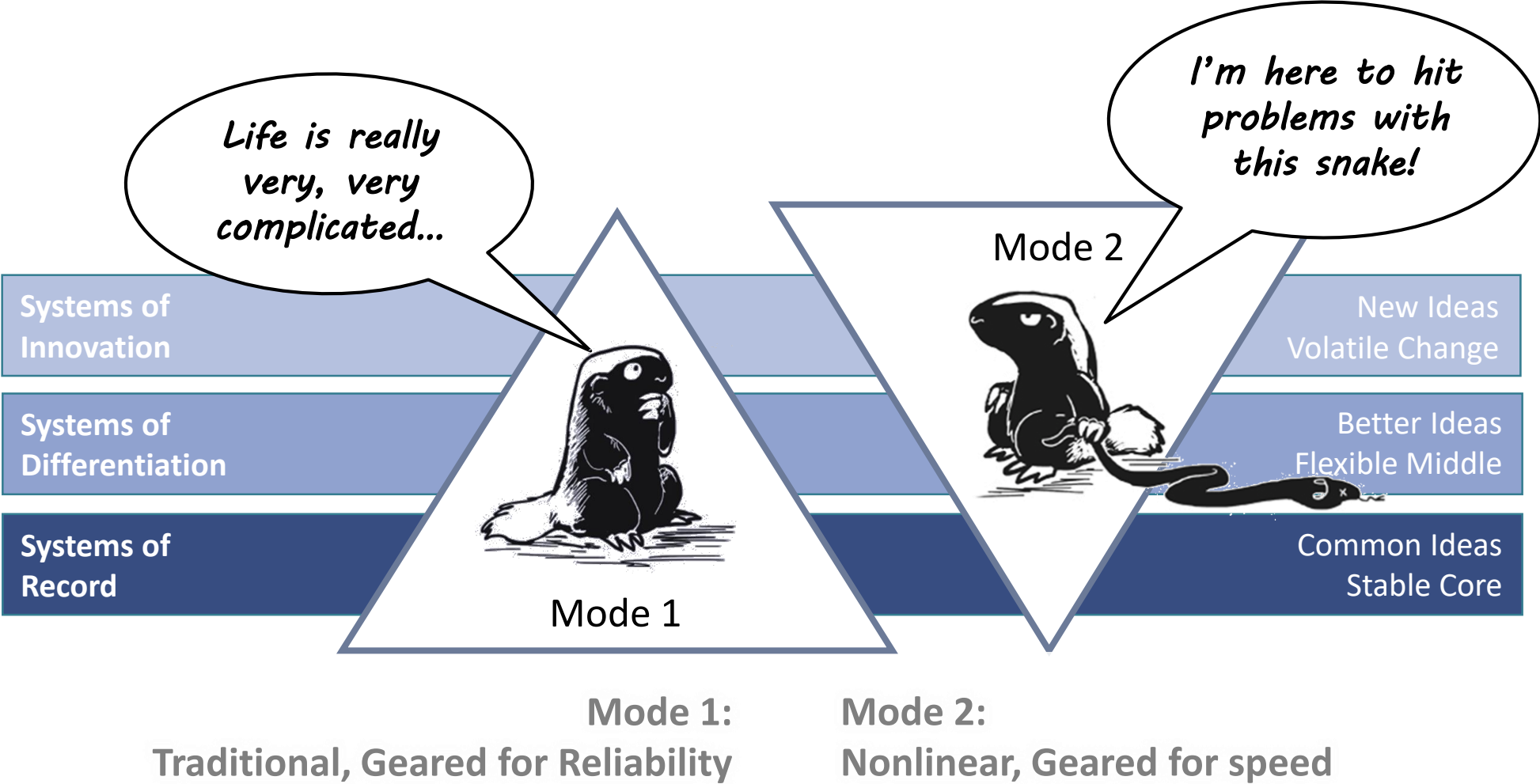


“Here be Dragons”

Humans coping with **new stuff** since 1362.

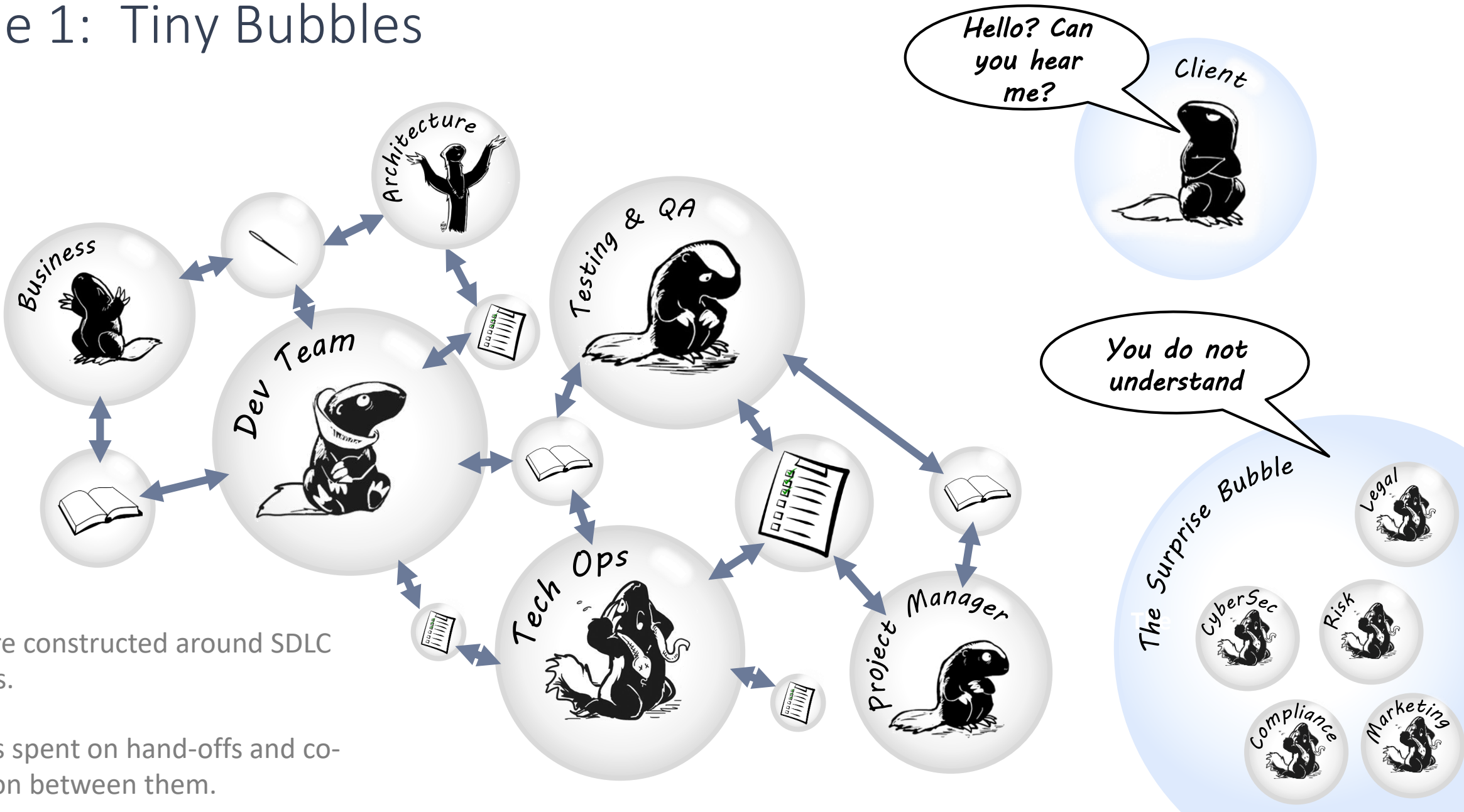
1362 BC... probably

Pace Layering and Bimodal IT...



Vintage 2012

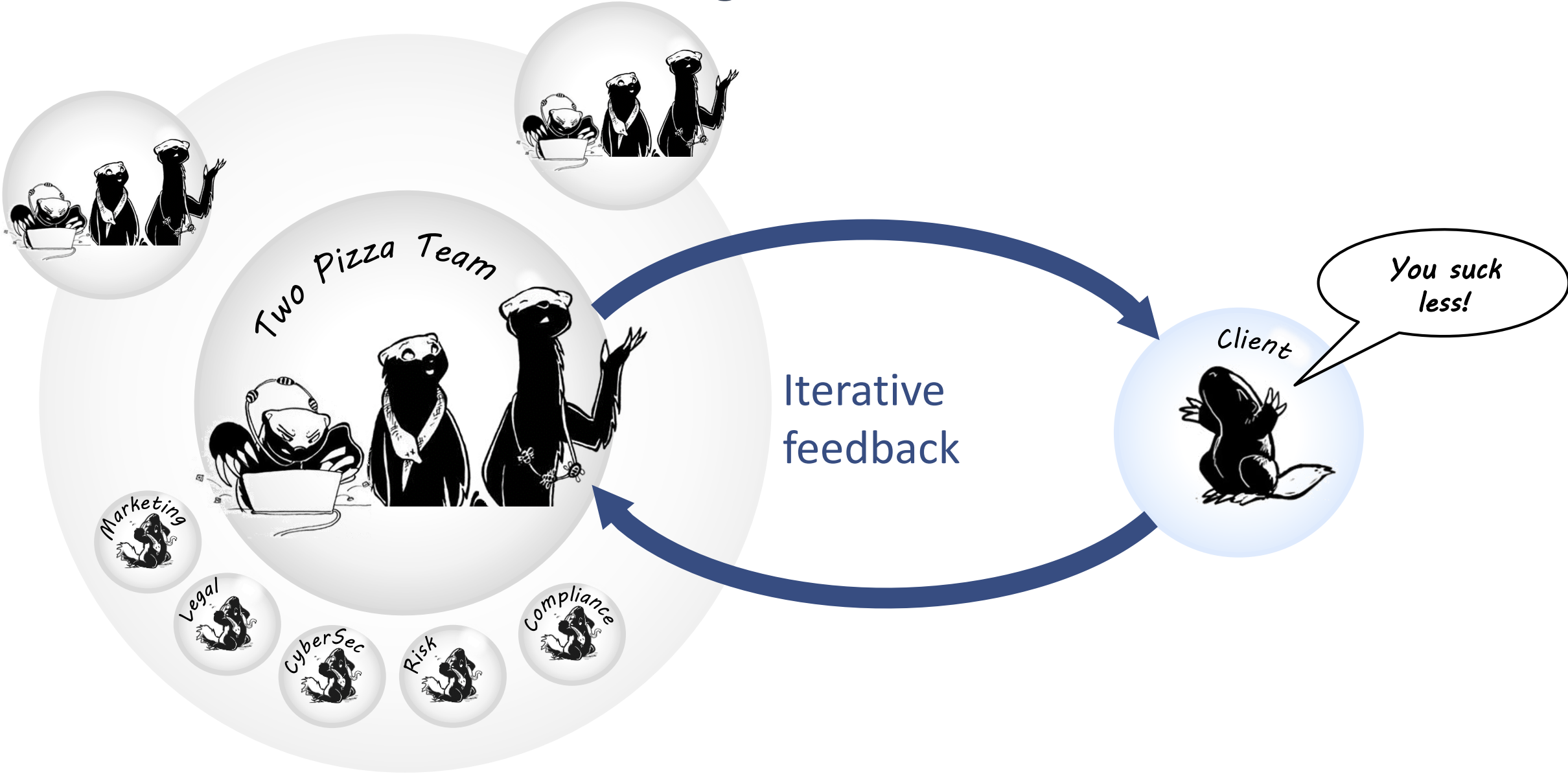
Mode 1: Tiny Bubbles



Teams are constructed around SDLC functions.

Energy is spent on hand-offs and coordination between them.

Mode 2: Two Pizzas or Five Engineers



Risk vs Agility - An Existential Crisis?



That's not the decision...

Pure bimodal is **vintage** thinking

Code Meets World: The Vicious Cycle as Complexity Grows



Code gets pushed to production without being “handed over” – causes instability, headache and heartache.

Feature build-up in pre-prod makes it impossible to separate a particular feature for test or deployment.

Ops teams become increasingly reluctant while Dev teams become increasingly desperate.

Segregation of “**dudies**” means that those who deploy have never seen the code and those who coded have never seen prod.

Code Meets World: The Cone of Shame



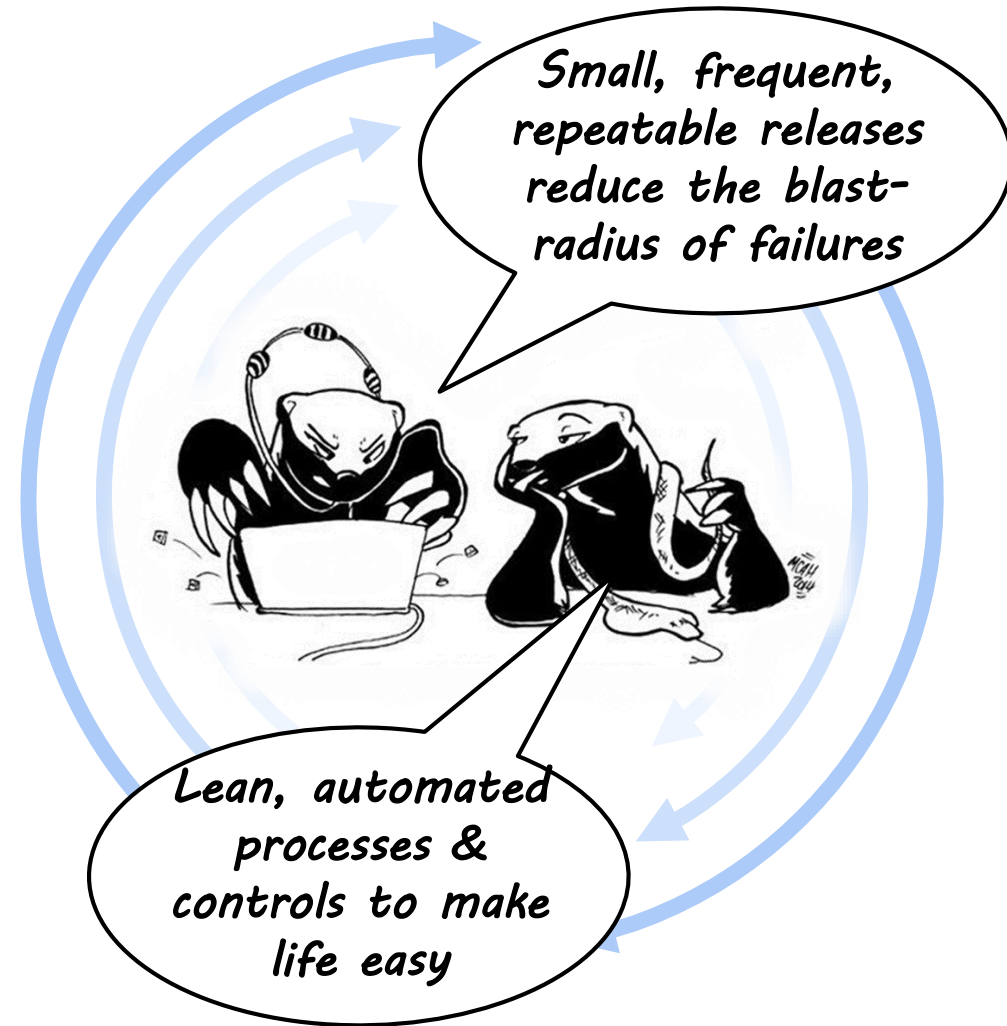
Releasing diverse stuff to production infrequently is fraught with risk – every deployment is different and complicated. It's called waterfall for a reason.

We need more governance!!!

No, you need **different** governance

Code Meets World: Small Frequent Releases...

Smaller releases and higher automation actually reduce your risk and increase your speed of delivery.

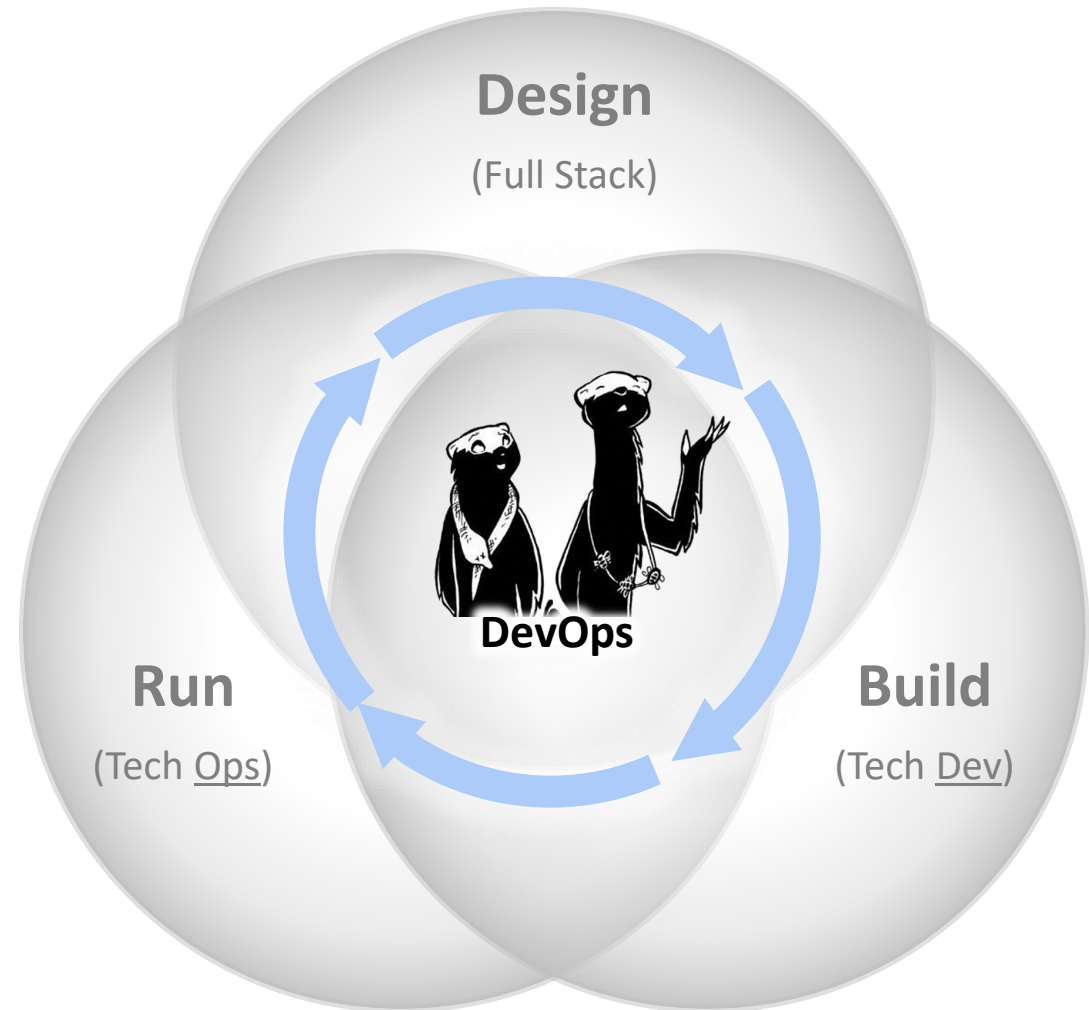


What is DevOps?

Instead of having separate functional bubbles, DevOps is a **cultural** movement which builds on top of agile.

At its core DevOps core philosophies expand on the principles of the Agile Manifesto...

- Communication & Collaboration
- Integration & Holistic thinking
- Automation & Self-service
- Useful, Timely Feedback
- Culture of Excellence & Trust



We need more servers!!!

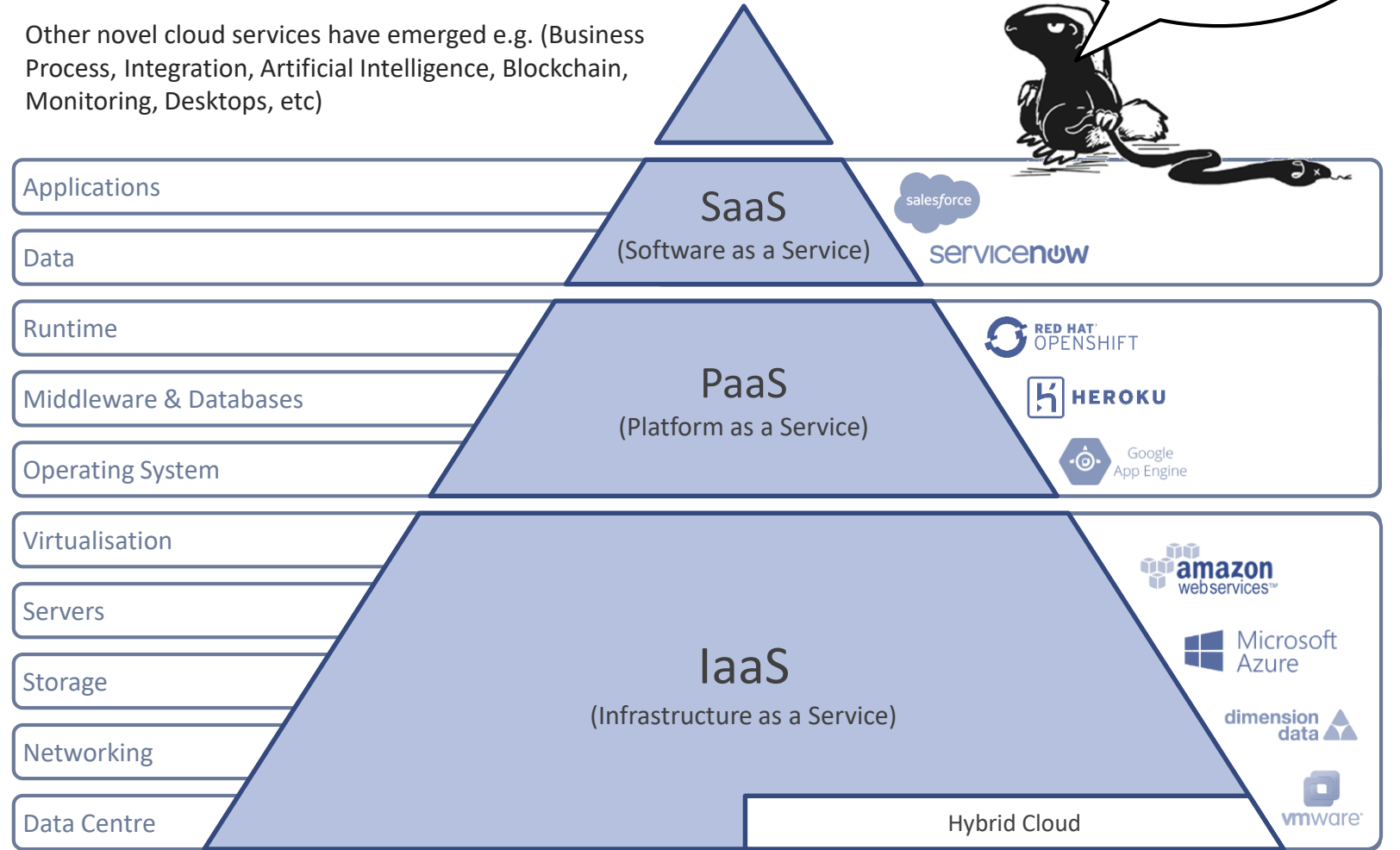
No, you need your tin to adapt as fast as the rest of you does.

So what is this cloud?

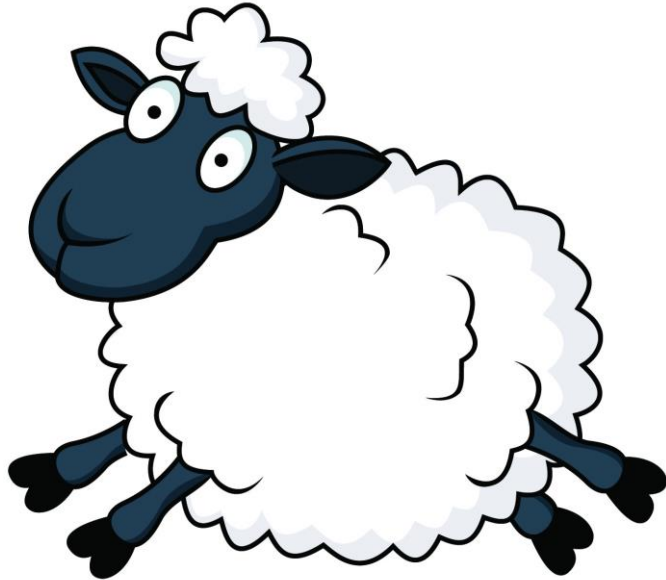


- This thinking is vintage 2007 (things have changed).
- The core of cloud is still technical infrastructure outsourcing
- There are a variety of different models

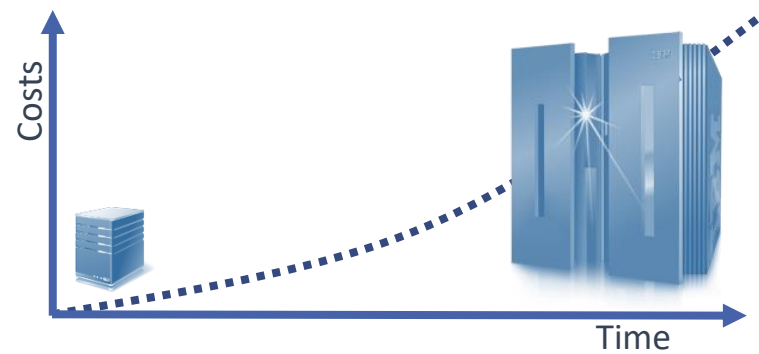
Other novel cloud services have emerged e.g. (Business Process, Integration, Artificial Intelligence, Blockchain, Monitoring, Desktops, etc)



Pets vs Cattle (Or Sheep)

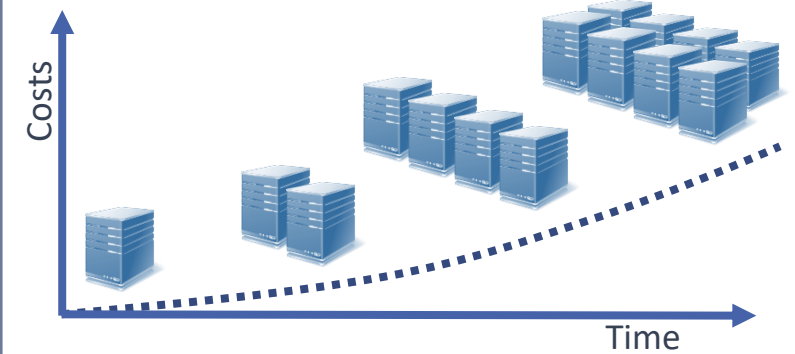


The Old Days (Pets)



- You buy bigger servers
- Each specially engineered
- Laboriously hand-reared (tickets)
- When sick they get special care
- Veterinarians on stand by
- You name your servers

Today (Cattle or Sheep)



- You buy more of the same
- Just a part of the herd
- Built fast and automated
- Sick ones get shot
- You have sheep dogs not vets
- You name your products

So your team is probably feeling a bit like this right now...

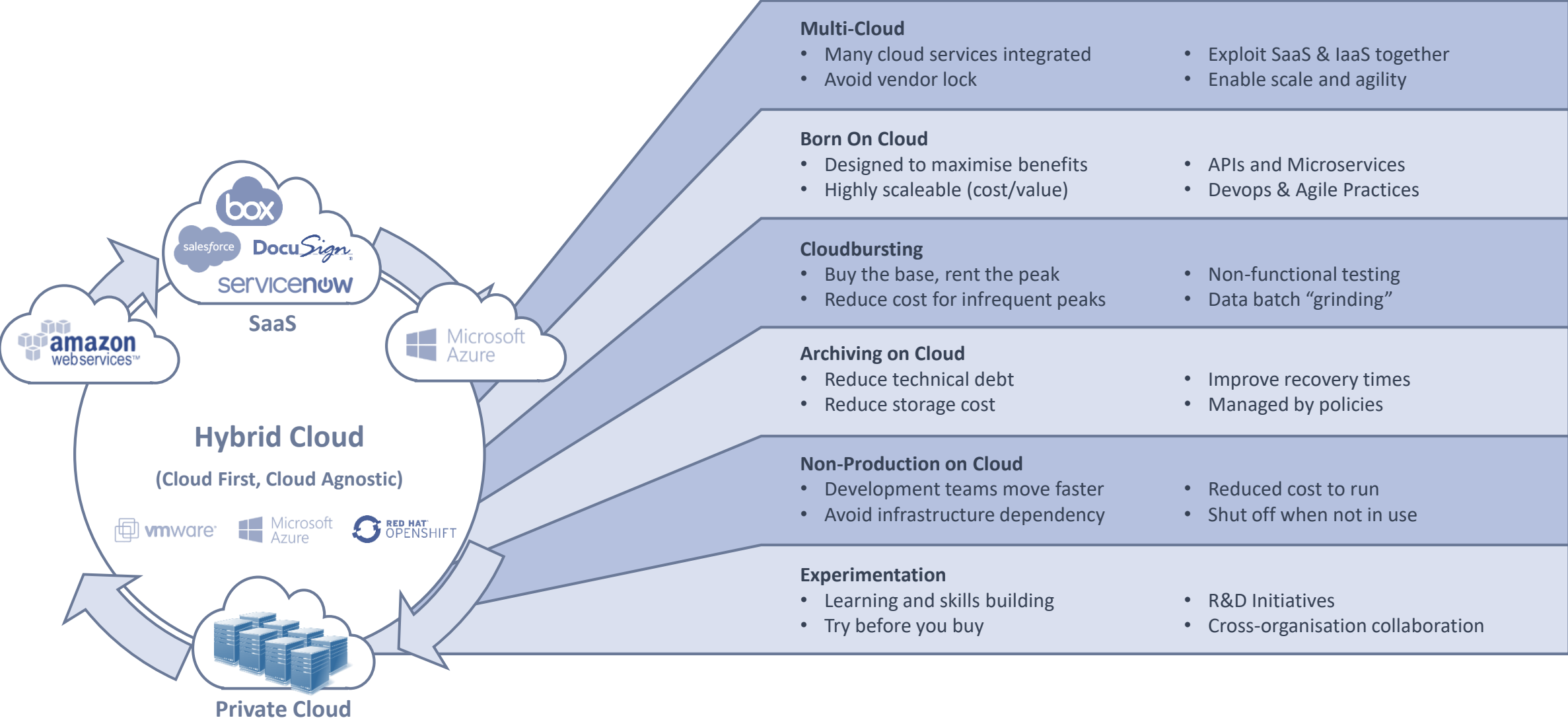
*Life is really
very, very
complicated...*



*How can I
possibly
decide?*



Rather than deciding (with some random vendor names)



Different Strokes (The Culture Shift)



- I have a checking things team
- I navigate many committees for approval.
- “The standard” is static and hidden
- I have pets and veterinarians
- I am trying to prove the value of EA
- I have an elaborate technology strategy
- Spend on following process (cost of control)
- I name applications and servers (transport)



- I automate things and I am my brother’s keeper
- The committees have delegates in the team.
- Standards are integrated, evolving and open
- I have sheep and sheep dogs
- Nobody is an EA...and everyone is an EA
- I have a strategy (tech is part of business)
- Spend on improving process (profit of control)
- I name business services (jobs to be done)

Obvious stuff I Learnt the Hard Way



- Nothing worth doing is easy **or known up front**. I don't have this all figured out yet.
- Culture eats strategy for breakfast! Make friends, wear jeans and consider swearing at work
- Agile coaching to get teams up-to-speed. Use your experienced teams to help scale.
- Automated build, test and deploy... mandatory to do but not dictated as to how.
- Don't over concentrate your change... run in parallel
- Scaling & portfolio planning – Roadmapping across projects rather than architecture in a project.
- Decent tools (PCs, VCs, Headsets, Webcams). Time lost setting up. Don't go to war unarmed!

So coming back to Adele



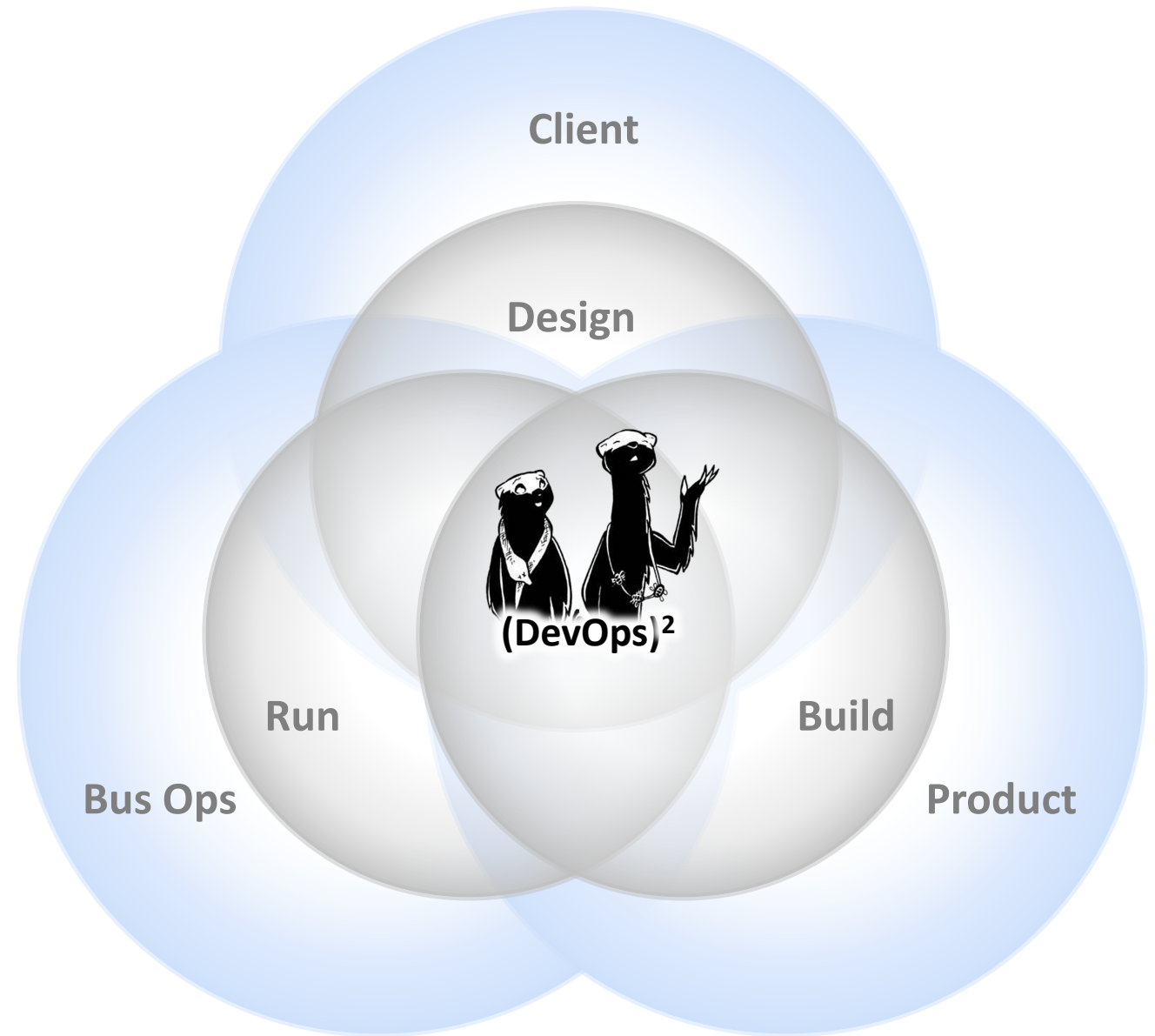
So this techy stuff is all well and good but how do I make this more business relevant and client focused?

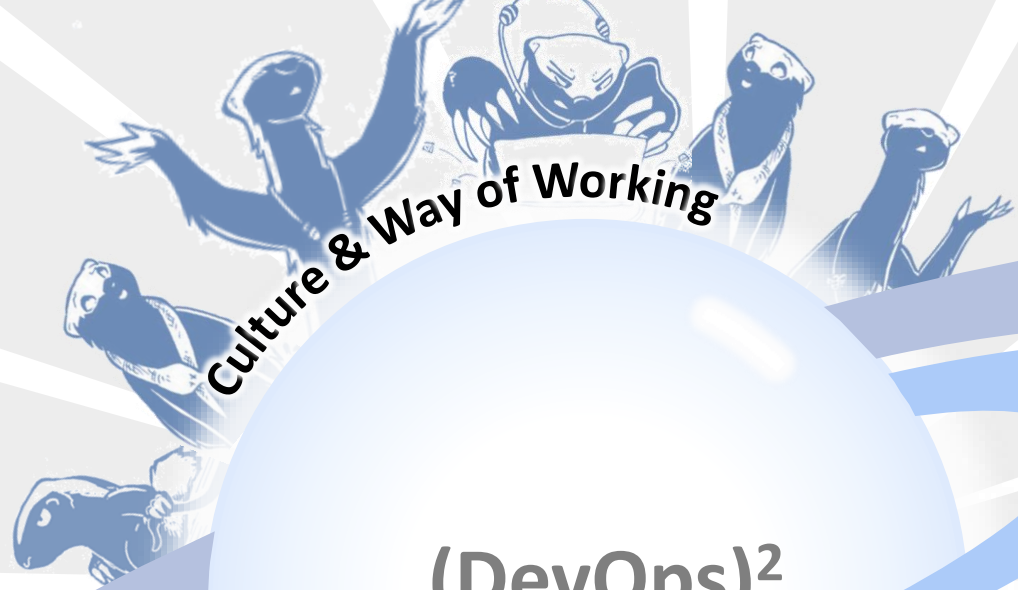
What's Next - (DevOps)2

To remain relevant, we need to deliver collaboratively across all areas of our an organisation.

Although DevOps has become very nerdy is not just about tech - “The Phoenix Project

Look towards Human Centred Design, Design Thinking, Cost of Delay Planning, Jobs to be Done... create a “Perfect Storm”





Culture & Way of Working

(DevOps)²

Client

Tools

Hybrid Cloud

Agile EA

TBA

A pretty slide to roll all this up...

Questions?