

# Right Abstraction

Yogi | @yogendra  
March 2019

# About Me - Yogi

- ❖ [yogendrarampuria@gmail.com](mailto:yogendrarampuria@gmail.com)
- ❖ @yogendra
- ❖ Programmer for Life
- ❖ Passionate about Productivity
- ❖ Work at Pivotal
- ❖ Live in Singapore
- ❖ Regular at Local Community (CF, Spring UG, GDG-SG, Kotlin, GCP UG, etc.)



What is Abstraction?

# Abstraction

the process of removing  
physical, spatial, or  
temporal details



“The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context”

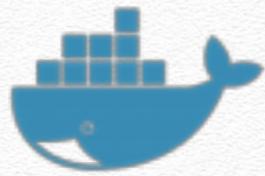
– John V. Guttag

# Challenges of Enterprises

# Variety of Workloads



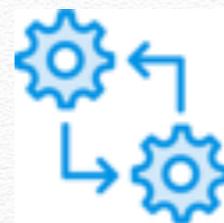
MICROSERVICES



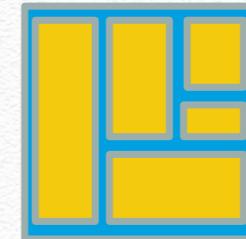
CONTAINERS



DATA SERVICES



Batches

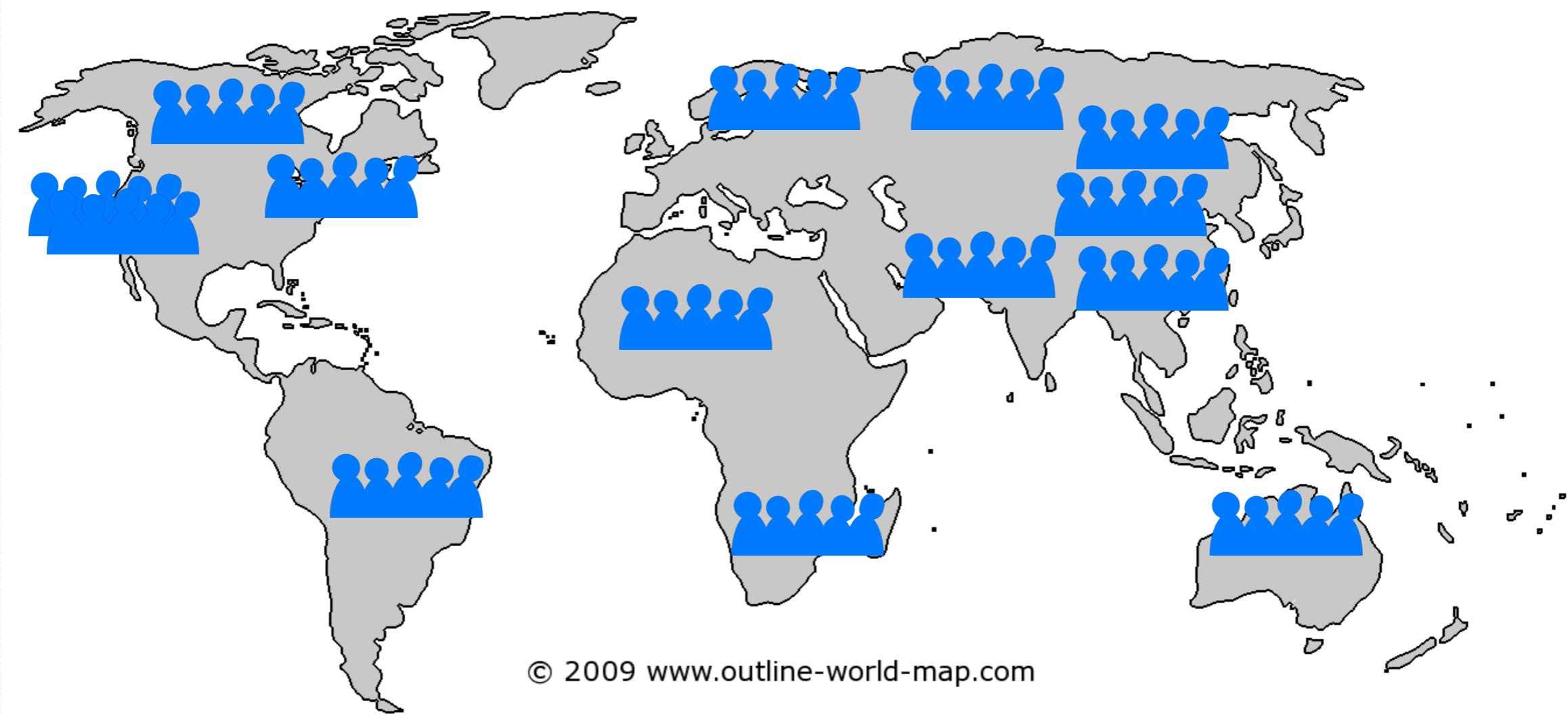


MONOLITHIC  
APPLICATIONS



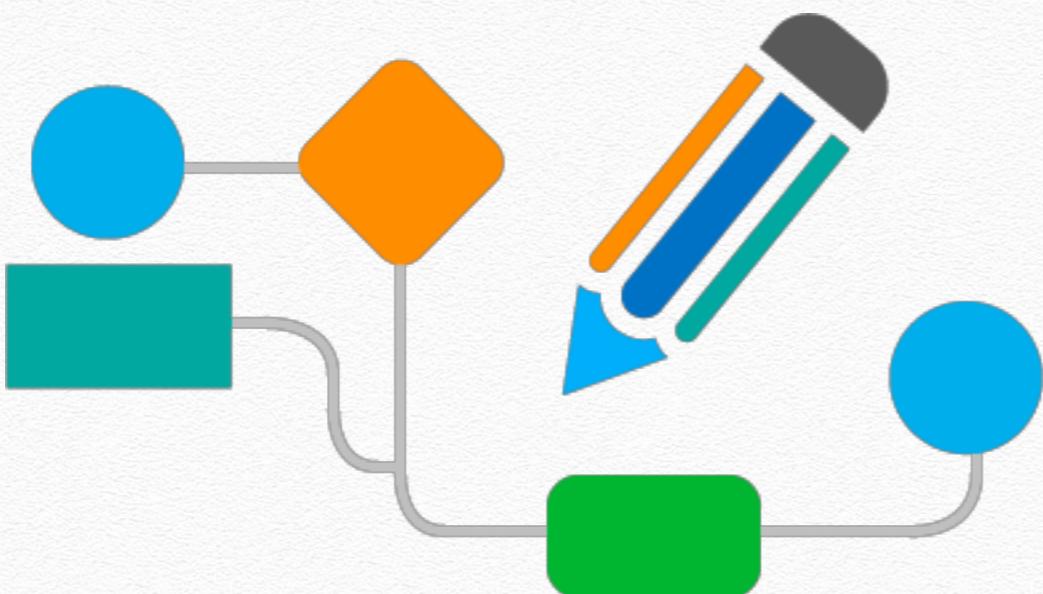
EVENT-DRIVEN  
FUNCTIONS

# Many Large Teams



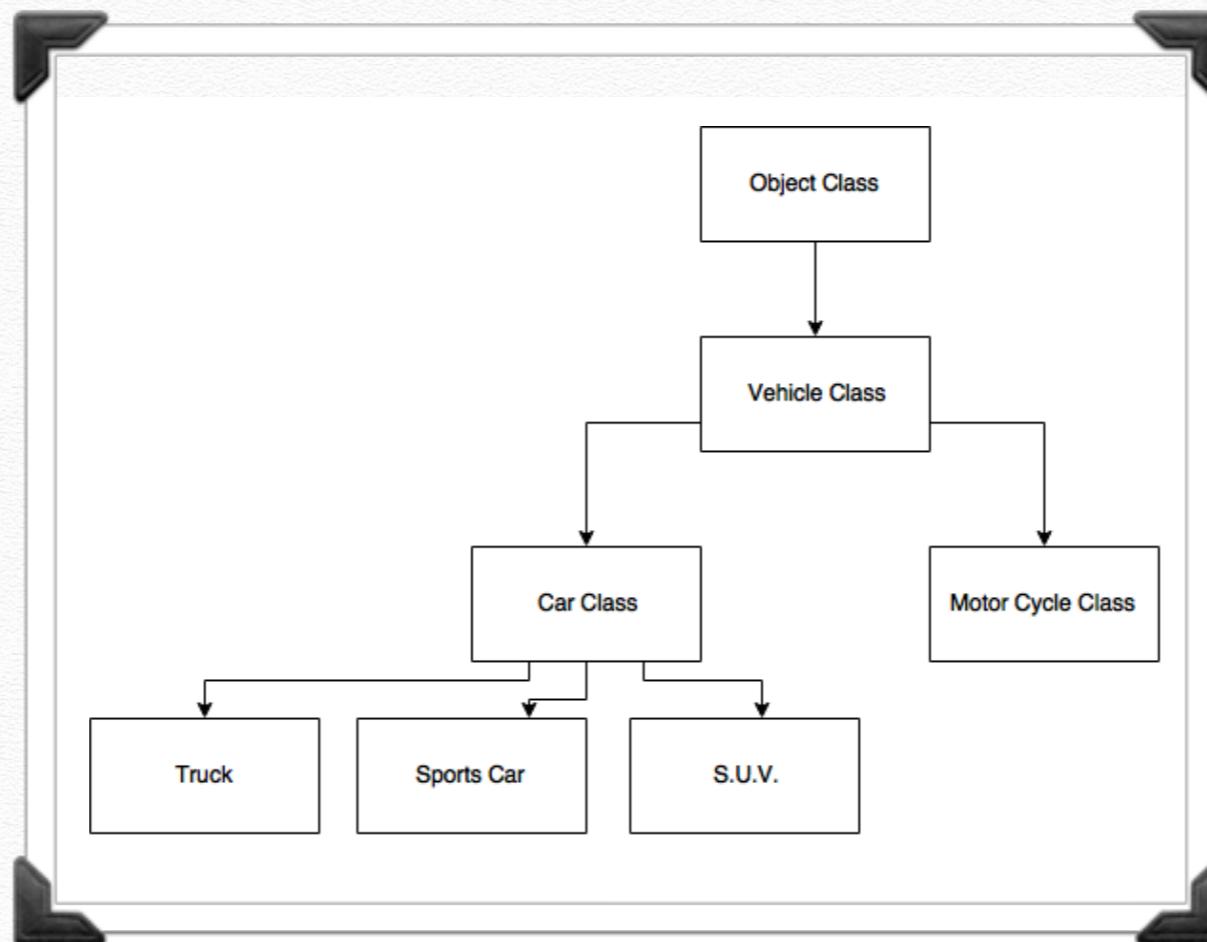
© 2009 www.outline-world-map.com

# Large Legacy Investment

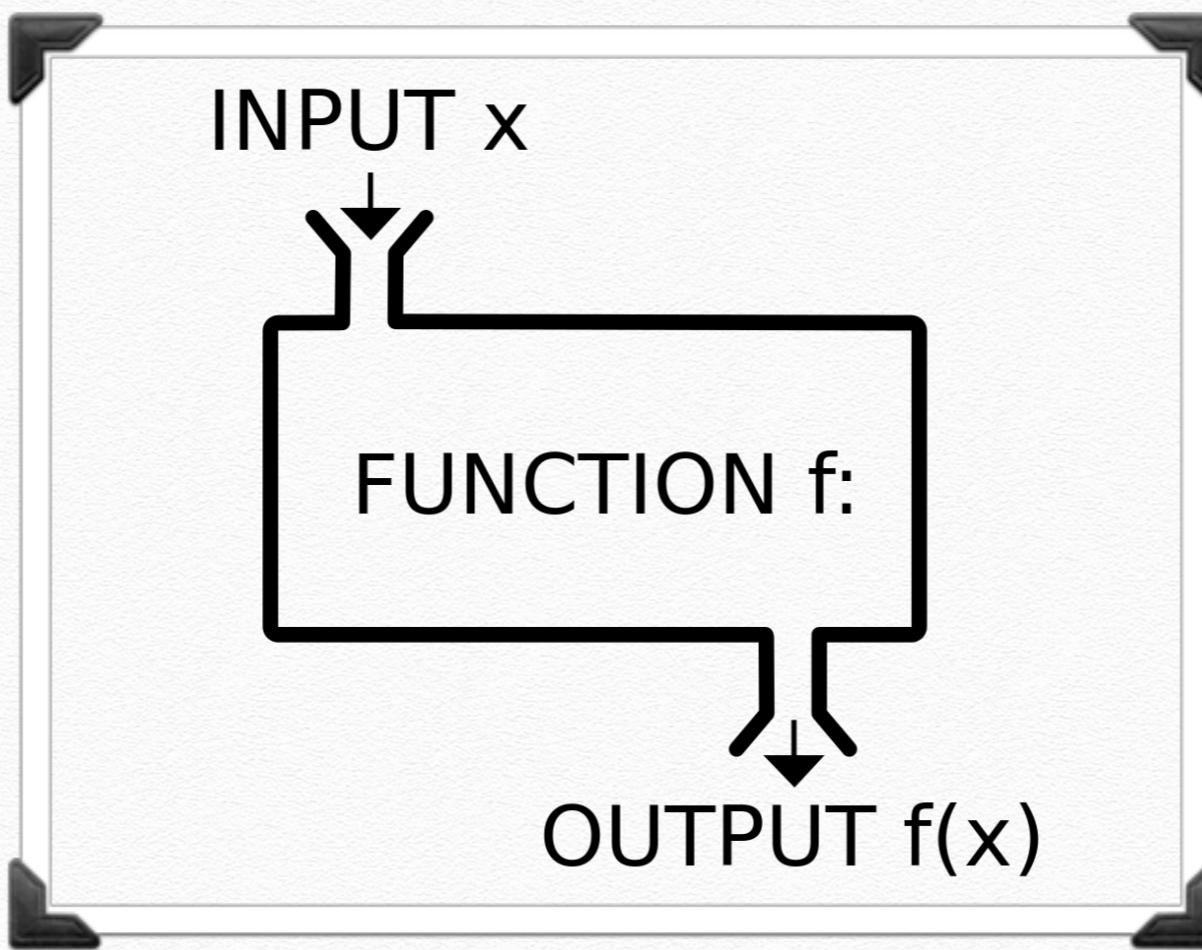


# Abstraction in Software Engineering

# Object Oriented Programming



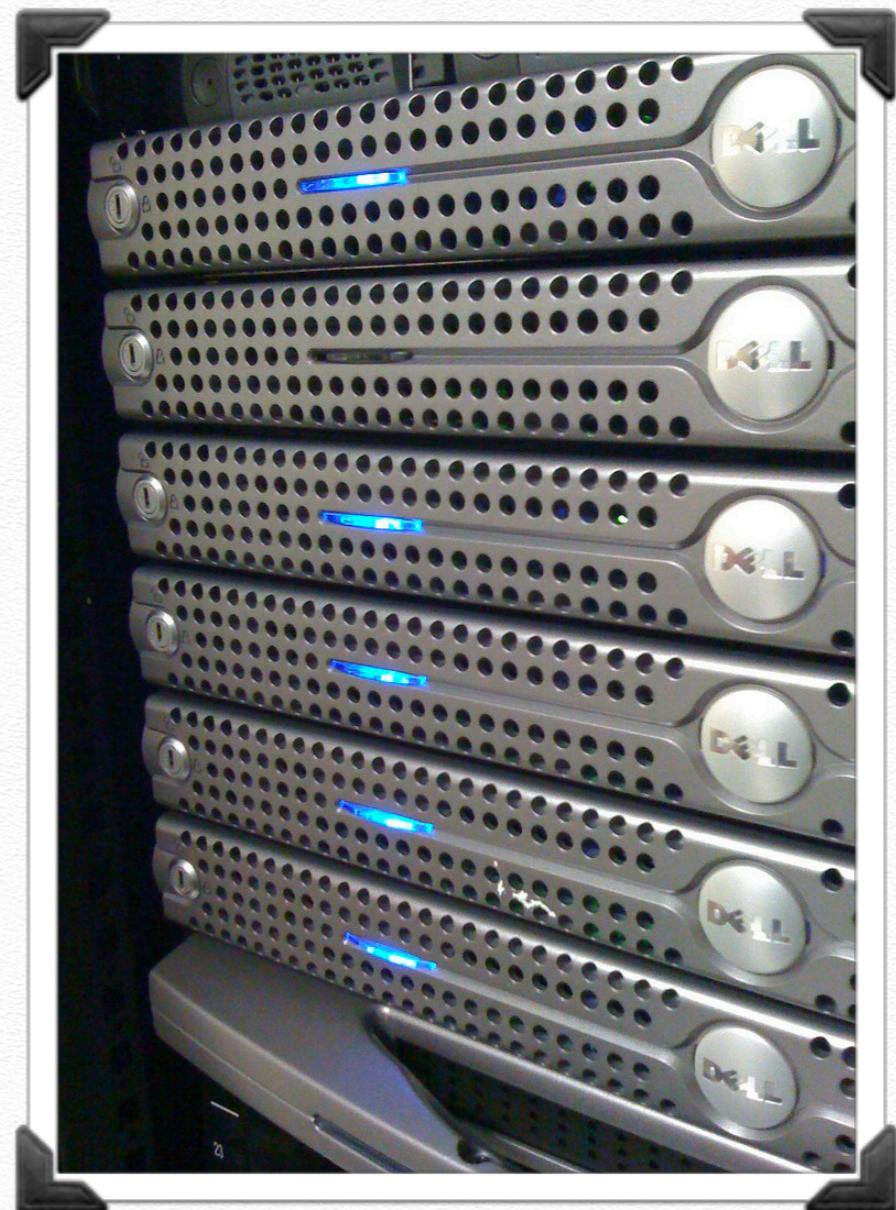
# Functional Programming

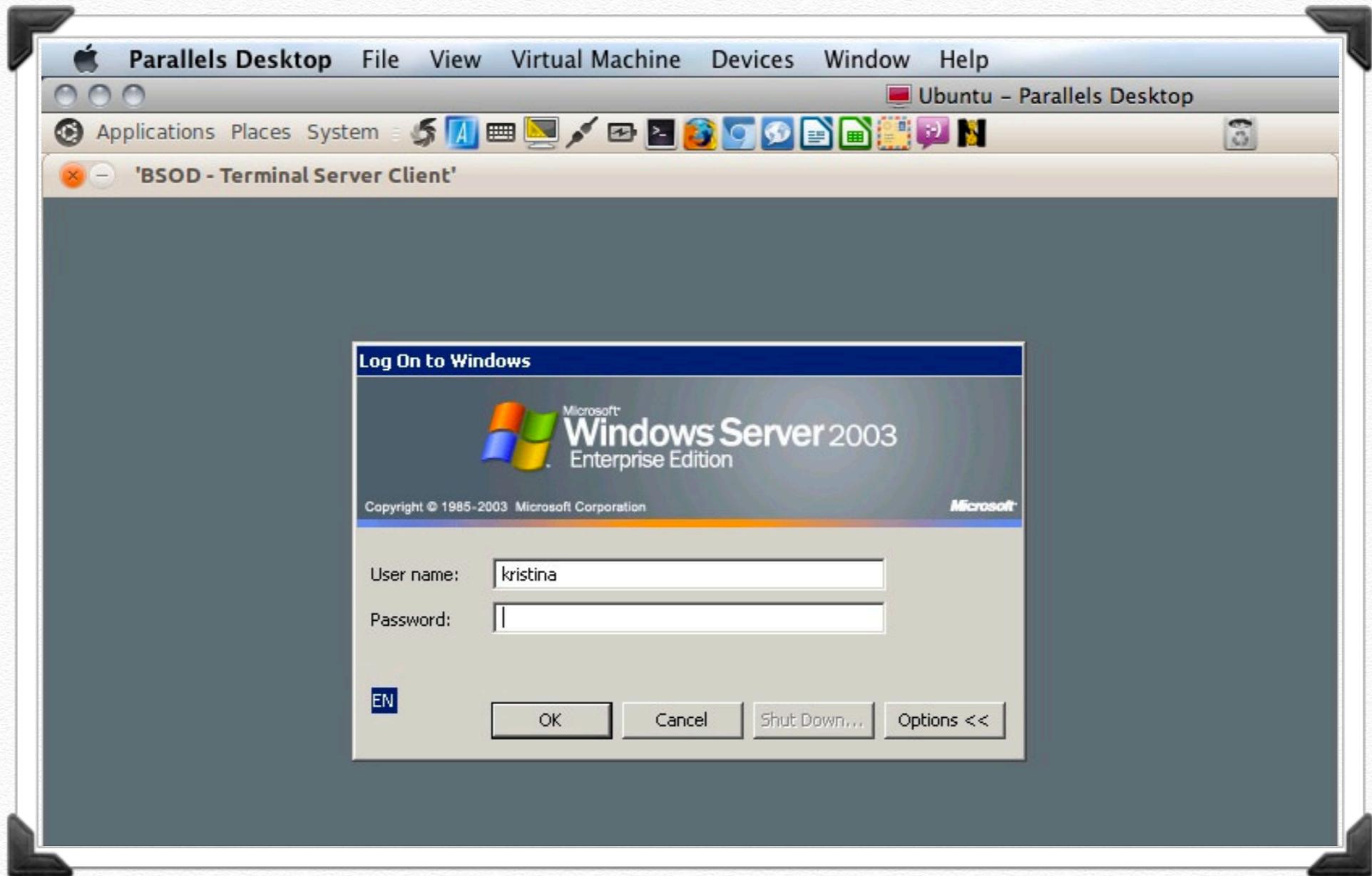


# Abstraction in Software Runtime

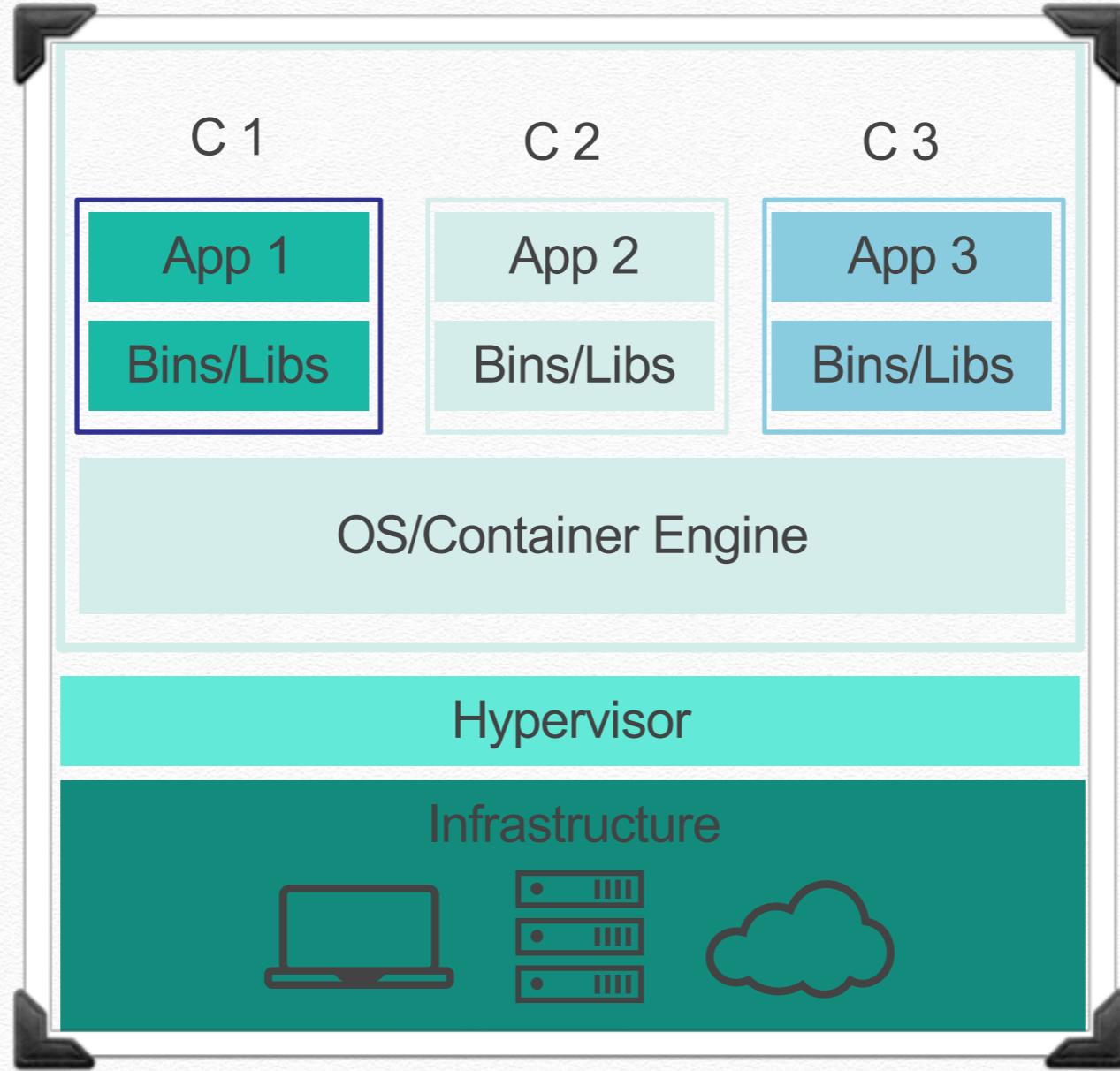
# Bare Metal

Hosts running in DC  
(or under the desk, next to  
coffee machine)





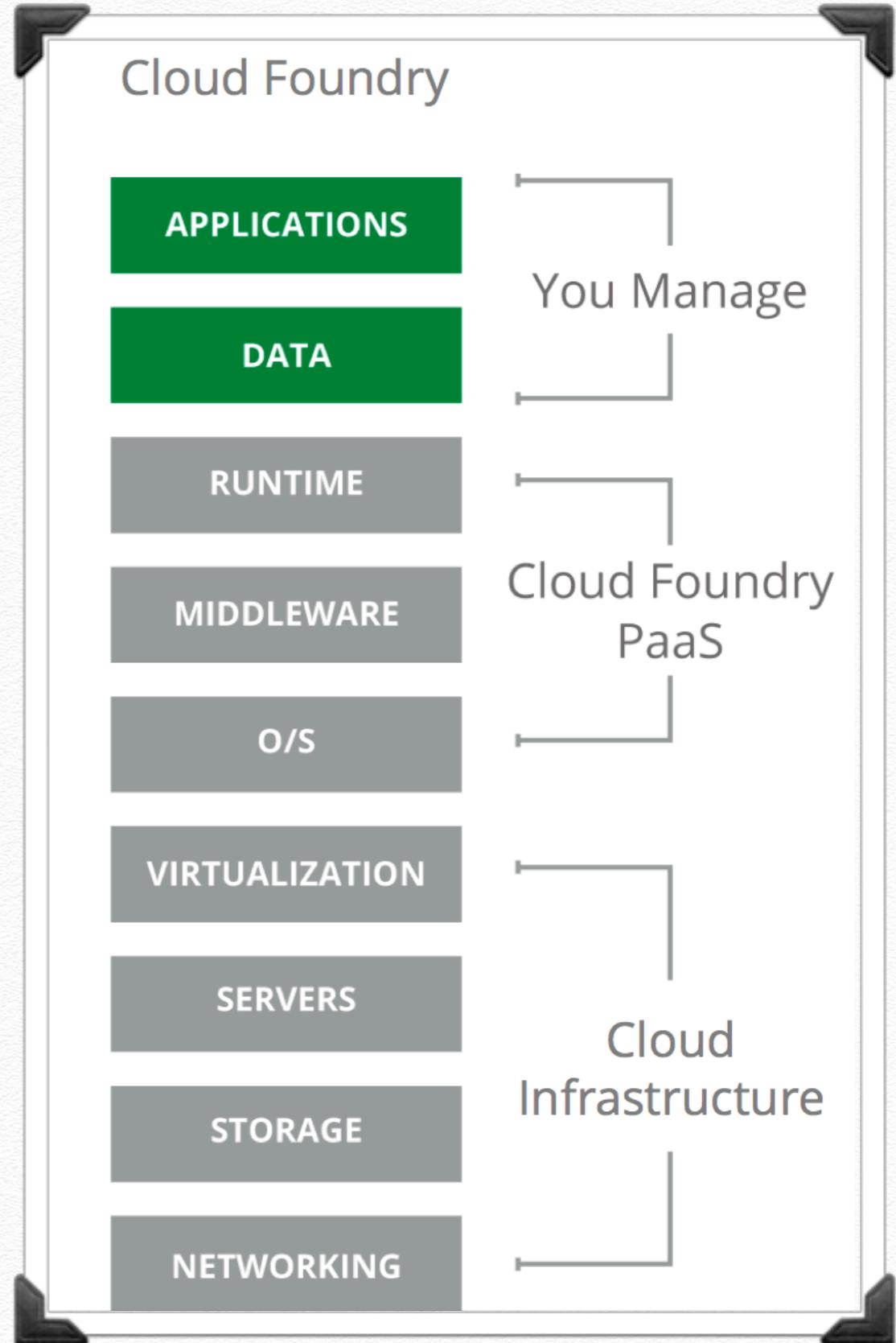
# Virtual Machine



# Containers

# Application

## Cloud Foundry, GAE



# Serverless

Knative, AWS Lambda, GCF



The screenshot shows the Spring Cloud Data Flow (SCDF) Streams interface. At the top, there's a navigation bar with links for Apps, Runtime, Streams (which is highlighted in green), Tasks, Jobs, Analytics, and About. Below the navigation bar, the title "Streams" is displayed, followed by a sub-instruction: "This section lists all the stream definitions and provides the ability to deploy/undeploy or destroy streams." There are two tabs: "Definitions" (selected) and "Create Stream". Under the "Definitions" tab, there are buttons for "EXPAND ALL" and "COLLAPSE ALL". Below these buttons, three status indicators are shown: "NO STREAM SELECTED TO DEPLOY" (green), "NO STREAM SELECTED TO UNDEPLOY" (green), and "NO STREAM SELECTED TO DESTROY" (red). To the right of these indicators are "Filter definitions" and a refresh icon. The main area displays a table of stream definitions. The columns are "Name" (with a dropdown arrow), "Definitions" (with a dropdown arrow), "Status" (with a question mark icon), and "Actions". One row is visible, representing the stream "Upper-Case-Stream" with the definition: "http --port=7171 | transform --expression=toUpperCase() | file --directory=c:/dataflow-output". The status is "deployed" and the actions include icons for info, stop, start, and delete. Below the table, a flow diagram shows the stream definition: "http" → "transform" (with a lambda symbol λ) → "file". A zoom control at the bottom right indicates "200%". At the bottom of the page, there are navigation links: "« Previous", a blue "1", and "Next »".

# Specialized Runtimes

## Spring Cloud Data Flow (SCDF), Workflow Runtimes, Rule Engines

# Summary

# Physical Host

- ❖ Very limited use cases
- ❖ Legacy workloads
- ❖ Example: Mainframes

# Virtual Machines

- ❖ Vendor Provided VM images
- ❖ System or Runtime Software
- ❖ Example: Virtual Appliances

# Containers

- ❖ Trending upwards
- ❖ COTS Application
- ❖ Data Services
- ❖ Re-platforming monoliths
- ❖ Example: Elastic Search, Cassandra, NLP Engines, etc.

# Application

- ❖ Specialized/Standardized Container Runtimes
- ❖ Just Run It
- ❖ 12-Factor Apps, Microservices
- ❖ Ideal for API, Microsites
- ❖ Example: Spring Boot Applications, REST Service, Frontends

# Serverless

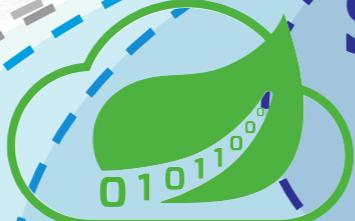
- ❖ Scale to Zero
- ❖ Ideal for event based systems
- ❖ Highly volatile traffic
- ❖ Example: Analytics processing, Order processing

# Specialized Runtime

- ❖ Brings the goodness of “Just Run It” and Serverless together
- ❖ Interconnectivity of services is a runtime concern and not coding concern
- ❖ Ideal for enterprises with 1000s of data driven applications
- ❖ Example: Customer service apps, Transaction processing apps, etc.

# Serverless Functions

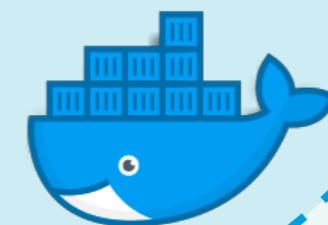
K<sup>n</sup>



# Application Platform



# Container Orchestrator



# IaaS



# Hardware

Thanks!  
Feedback and Questions?  
Always welcome

Reach Me @yogendra