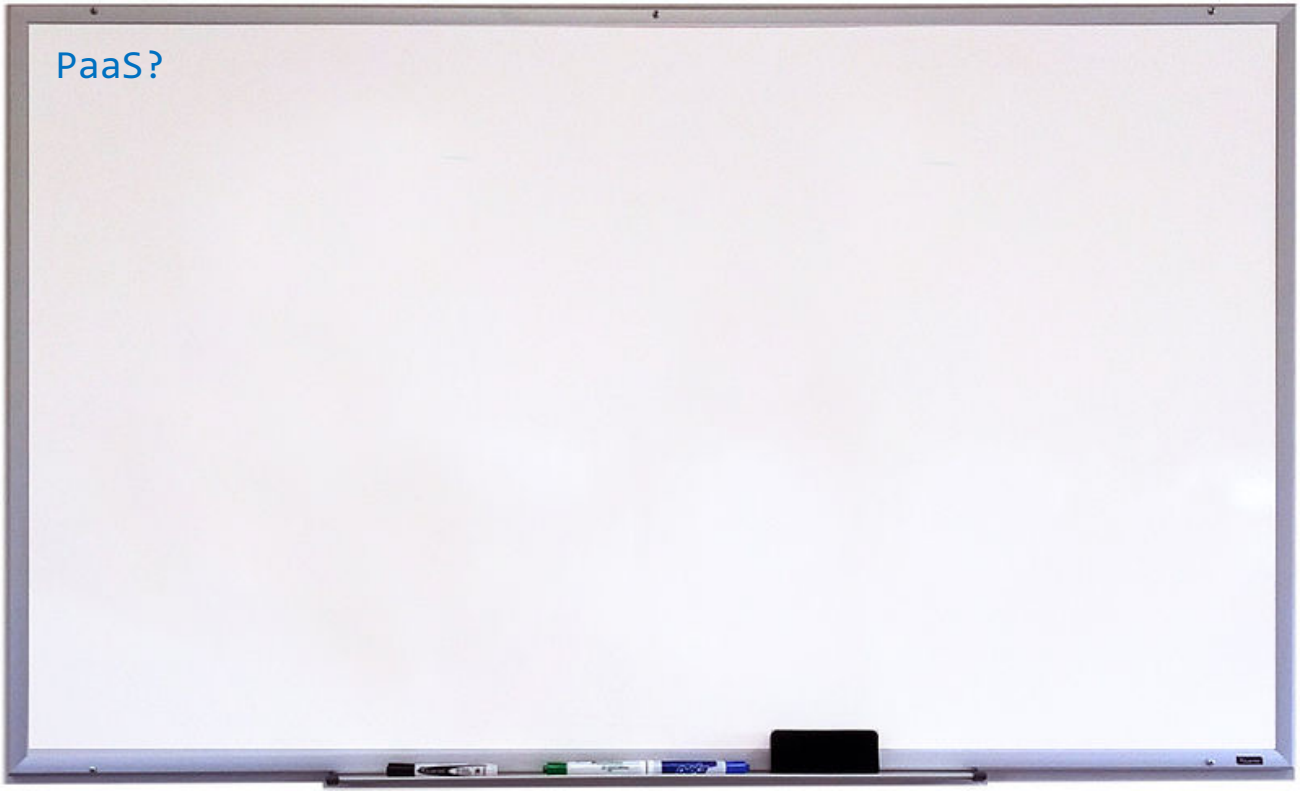
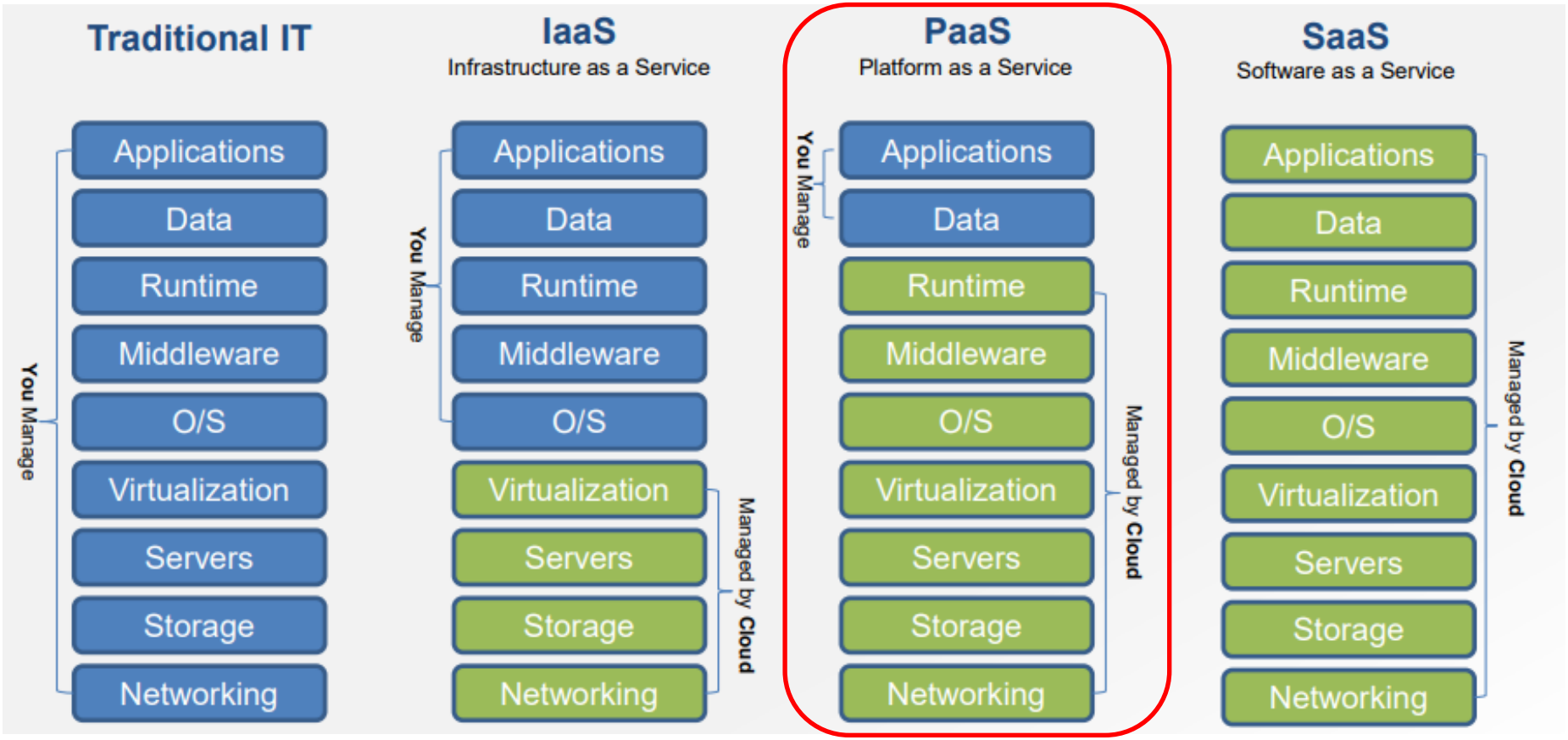
PaaS

Antonio Brogi

Department of Computer Science
University of Pisa

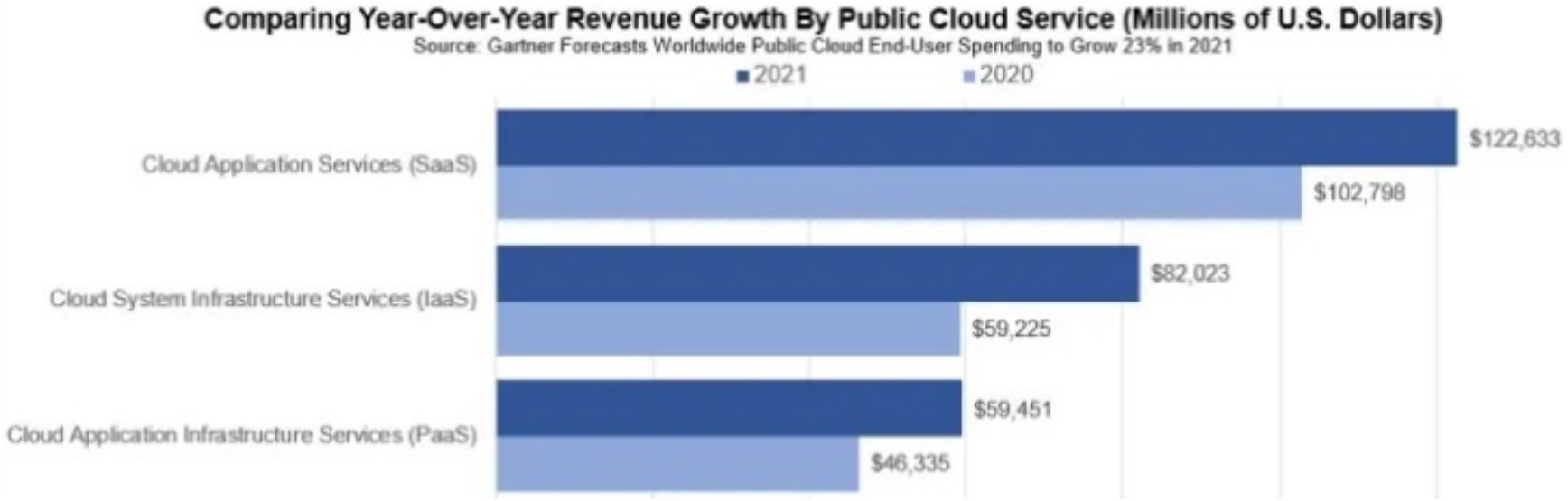

PaaS?


Traditional IT
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You Manage
IaaS
Infrastructure as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You Manage
Managed by Cloud
PaaS
Platform as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You Manage
Managed by Cloud
SaaS
Software as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Managed by Cloud

What is PaaS?

https://www.youtube.com/watch?v=6L1eQe5E1K4

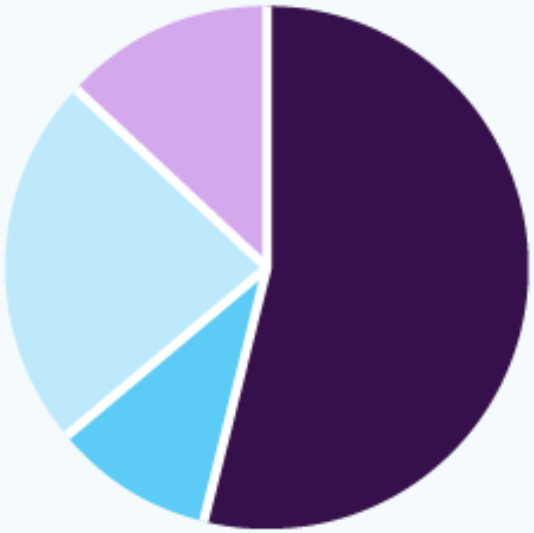
- Third-party providers deliver HW and SW tools for application development
- User just provides application and data
- Advantages
  - Decreased infrastructure management
  - Automated mainteinance
  - Easier load balancing, scaling, distribution of services
  - Easier adoption of new offerings and technologies
- Risks
  - Service availability
  - Vendor lock-in
  - Internal changes to PaaS



[Gartner (April 2021)]

Global Paas Market

share, by offering, 2021 (%)



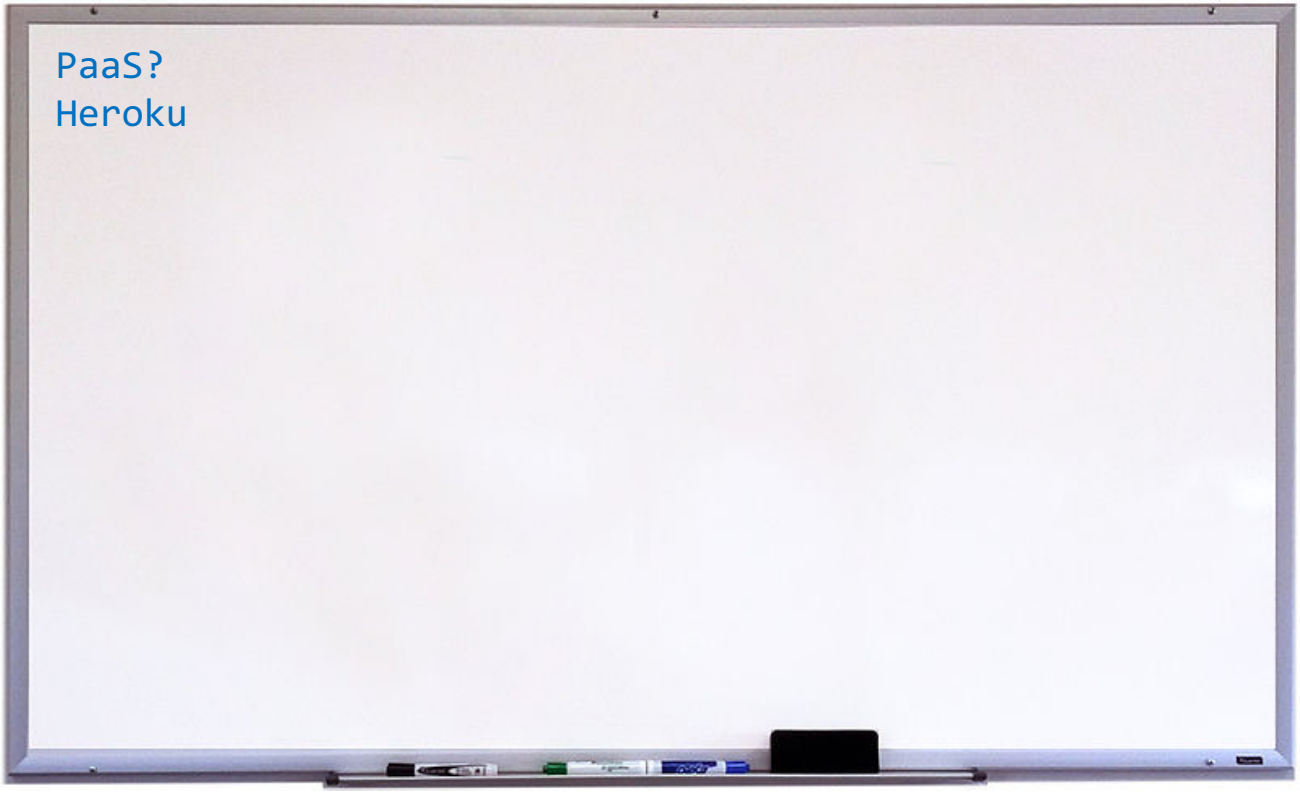
Application Development · Debugging processes · Testing · Deployment Tools

GRAND VIEW RESEARCH

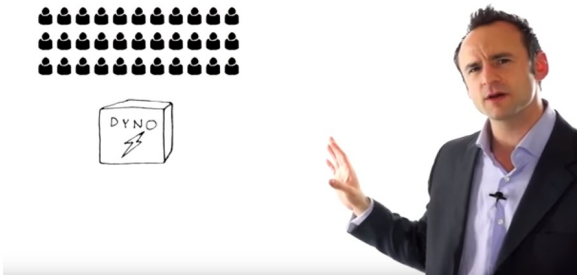
$60.1B

Global Market Size, 2021

Source: www.grandviewresearch.com


PaaS?
Heroku

What is Heroku? (by analogy)



Bob and Jim
- need lot of time to produce foods for occasional events
- often produce too much or too little food
- can only make hamburgers and pasta salads
- want to improve their business

They hear of a “cloud catering company” that “wows” people
- they have magical shelfs in their kitchens (any ingredient, any quantity, even finished goods)
- they can focus on using their imagination to create many different types of delicious food
- they ship recipies to factories -that can scale to meet any demand!

Heroku is similar to that “cloud catering company”
- can build with different languages
- many “add-on”s that are easy to plug in
- dynos for scaling up and down arbitrarily

You can focus on creating delicious applications!

Heroku

Heroku is a cloud platform based on a managed container system, with integrated data services and a powerful ecosystem, for deploying and running modern apps

(Born in 2007, acquired by Salesforce in 2010 for 212 MUSD)

Dynos

The Heroku Platform uses containers (“dynos”) to run and scale all Heroku apps

Dynos are isolated, virtualized Linux containers designed to execute code based on a user-specified command

- app can scale to any specified number of dynos based on its resource demands
- easy for user to scale and manage number, size, and type of dynos for app

Deploying to dynos makes it easy to build and run flexible, scalable apps - freeing user from managing infrastructure

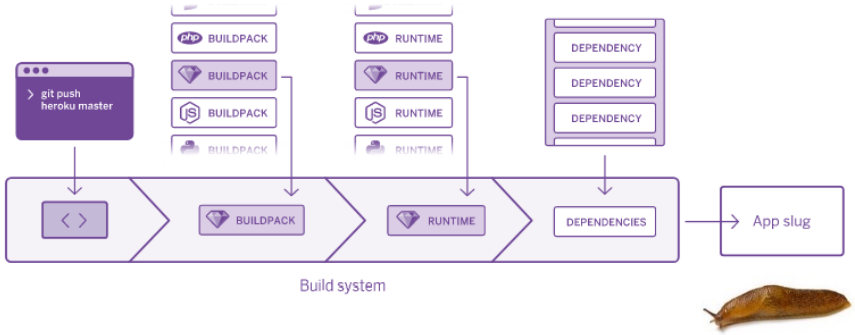
Buildtime

To deploy an app, Heroku needs only three things from the developer:

- source code
- a list of dependencies
- a “Procfile” (text file indicating which command to use to start the code running)

The automated build system

- receives your code
- fetches a buildpack, language runtime, and code dependencies
- produces a slug - a bundle of source, dependencies, runtime, output, that is injected into a dyno to run your app



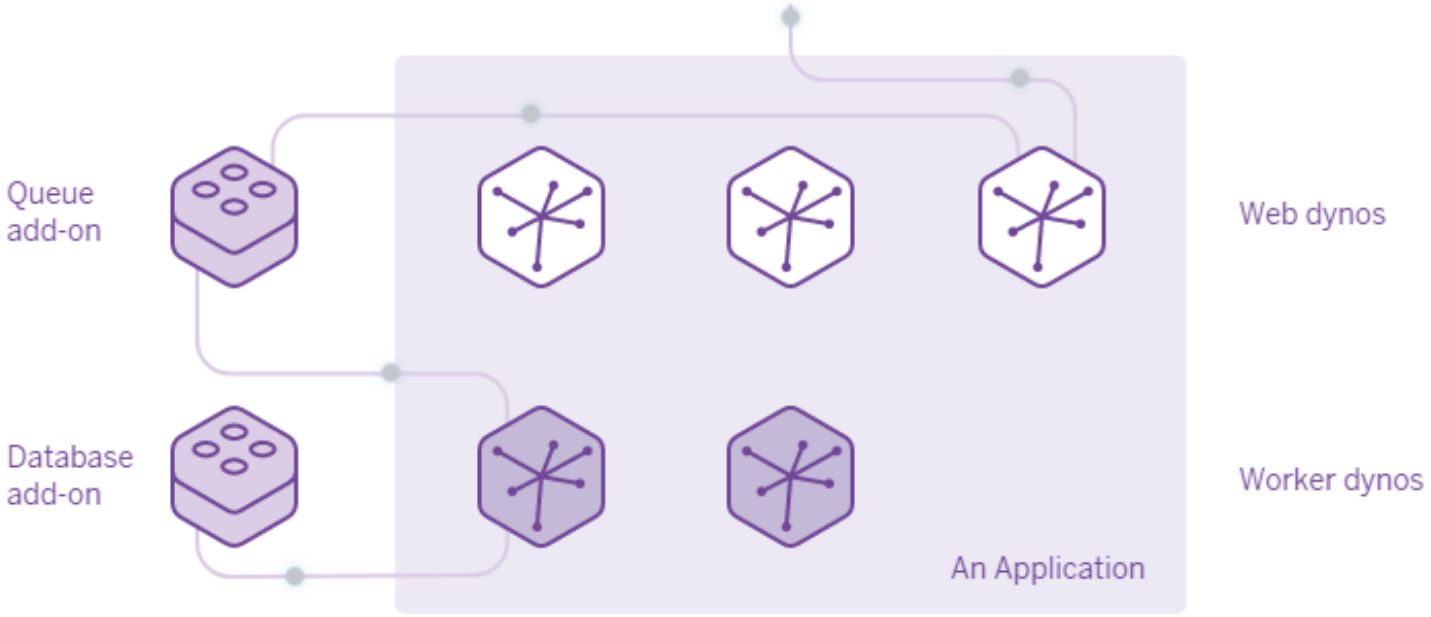
The final component needed to run app is the OS – on Heroku called the "stack", an Ubuntu operating system image maintained by Heroku

Runtime

When you deploy or scale your app, Heroku will automatically create one or more dynos, each loaded with the same stack and slug representing your app

Heroku's Dyno Manager then executes the command you provided in your configuration file to start your application running on Heroku

Dynos (example)



- Application receives request
- Request is delivered to random Web dyno
- Request is placed in queue
- Worker dyno picks up request and does the work, can persist result in database

Dyno types

Heroku allows developers to fine-tune their app's runtime resources by choosing different types of dynos

Dyno Type	Memory (RAM)	CPU Share	Compute	Dedicated	Sleeps
free	512 MB	1x	1x-4x	no	yes
hobby	512 MB	1x	1x-4x	no	no
standard-1x	512 MB	1x	1x-4x	no	no
standard-2x	1024 MB	2x	4x-8x	no	no
performance-m	2.5 GB	100%	12x	yes	no
performance-l	14 GB	100%	50x	yes	no

Standard dynos support scalability
Performance dynos support scalability and autoscaling

Dyno Type	Price per dyno/month
free	$0
hobby	$7
standard-1x	$25
standard-2x	$50
performance-m	$250
performance-l	$500

Heroku Add-ons

150+ 3rd party cloud services that developers can use to immediately extend their apps with a range of functionality such as data stores, logging, monitoring and more

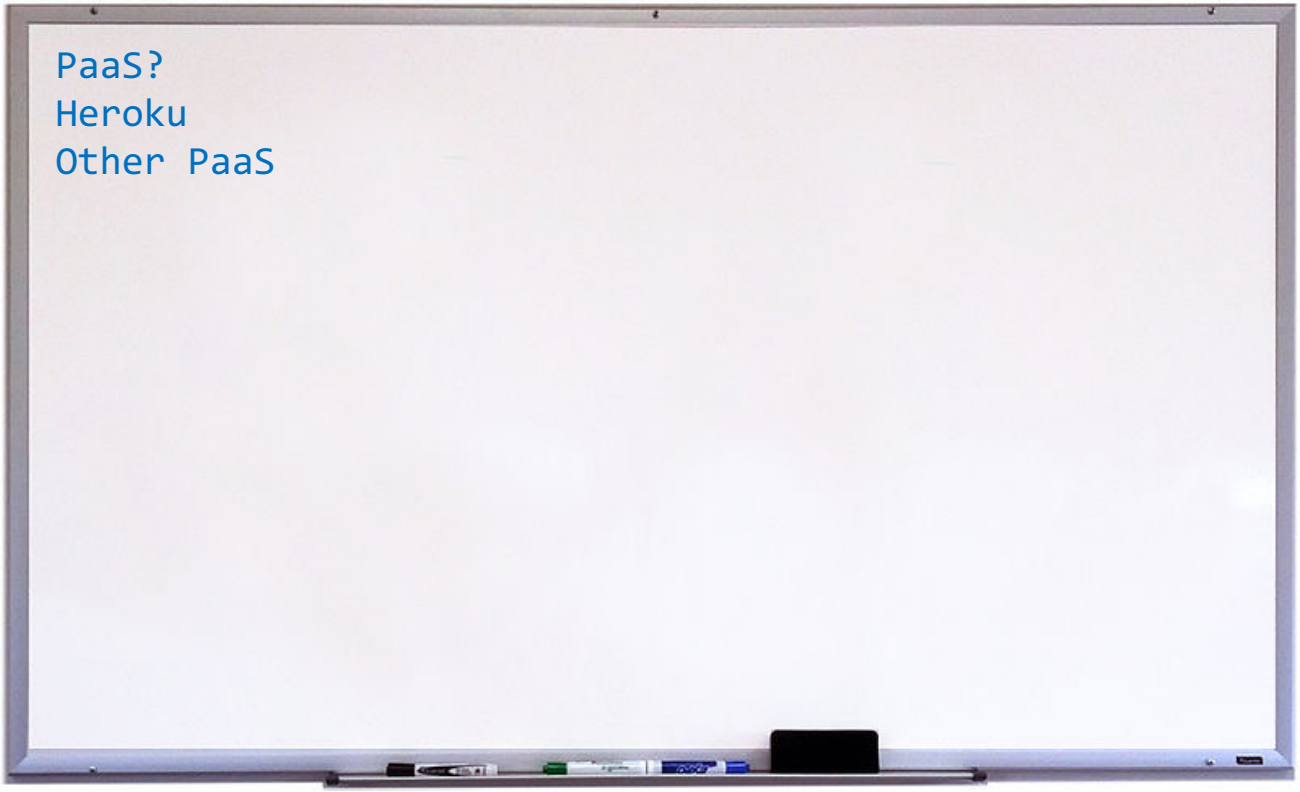
ADD-ON CATEGORIES

- Data Stores
- Data Store Utilities
- Monitoring
- Logging
- Email/SMS
- Caching
- Errors and Exceptions
- Content Management
- Search
- Metrics and Analytics
- Testing
- Messaging and Queueing
- Network Services
- Alerts and Notifications
- User Management
- Development Tools
- Security
- Dynos
- Content
- Document Processing
- Image Processing
- Video Processing
- Continuous Integration and Delivery
- Utilities

Heroku Add-ons

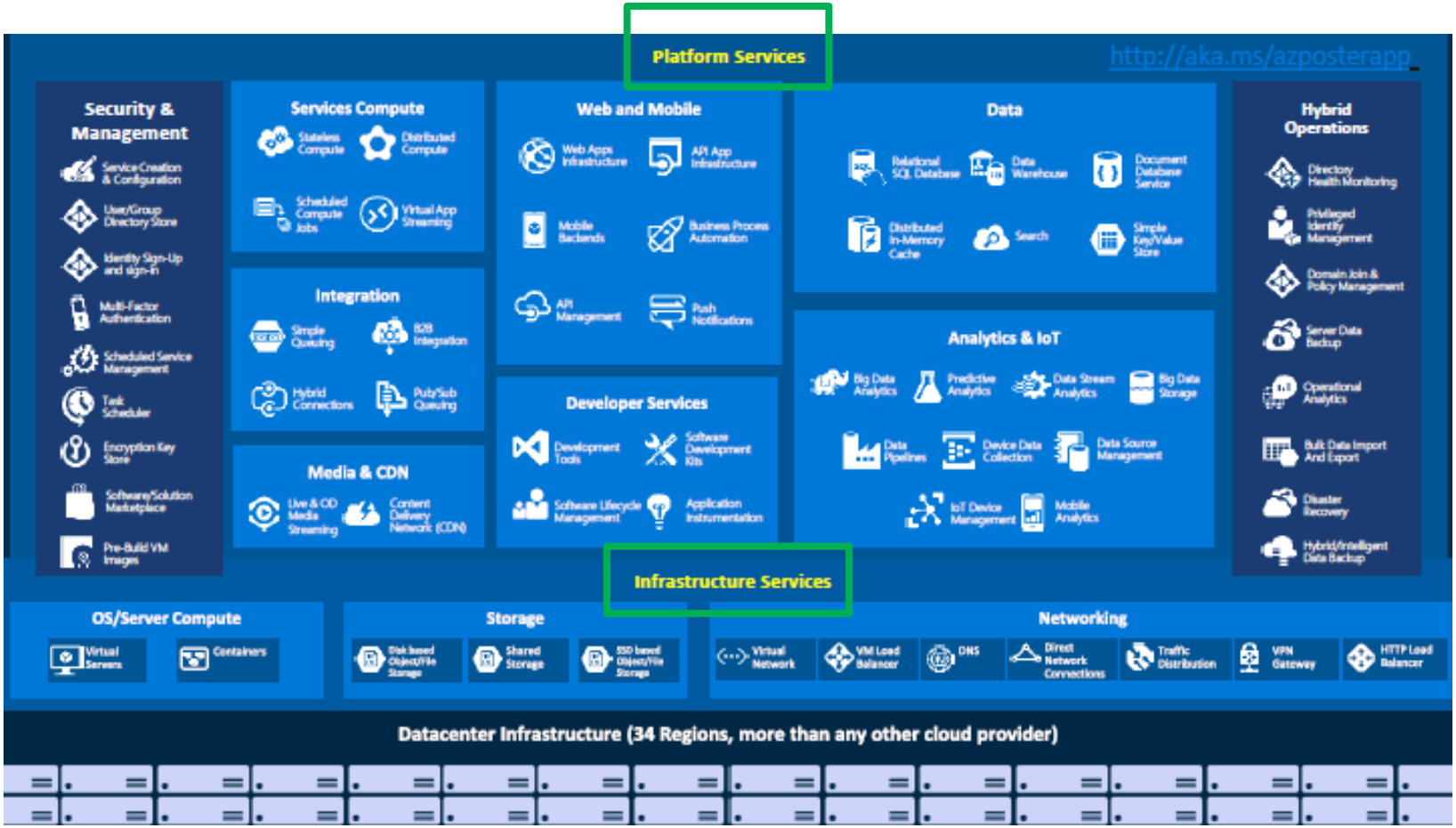
Monday, 5:50pm

We need to add some monitoring capability to our product prototype for the demo scheduled for Wednesday morning


PaaS?
Heroku
Other PaaS

Tour of Microsoft Azure

https://www.youtube.com/watch?v=0d1OO79brYY

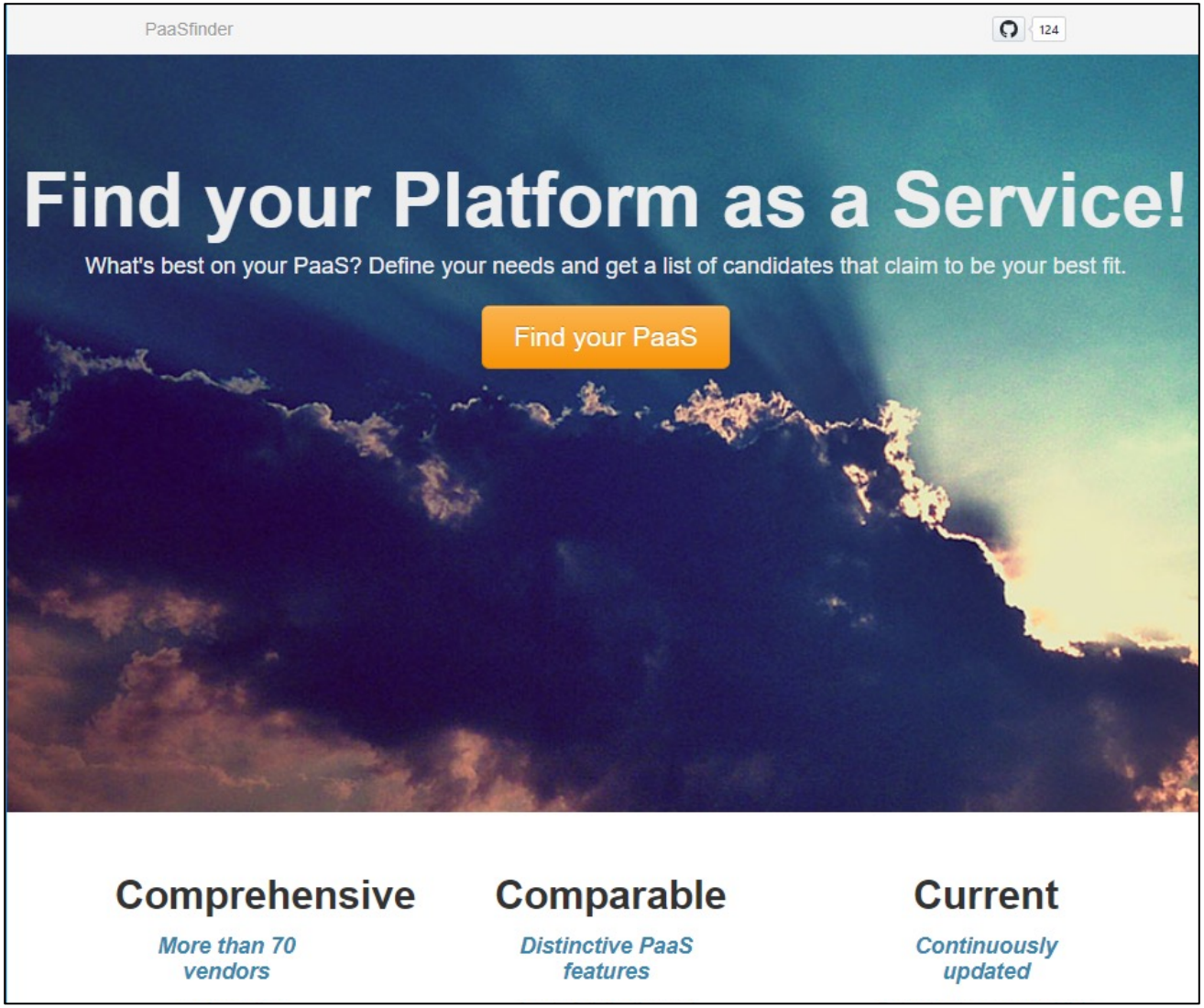

Platform Services
http://aka.ms/azposterapp
Security & Management
Service Creation & Configuration
User/Group Directory Store
Identity Sign-Up and sign-in
Multi-Factor Authentication
Scheduled Service Management
Task Scheduler
Encryption Key Store
Software/Solution Marketplace
Pre-Build VM Images
Services Compute
Stateless Compute
Distributed Compute
Scheduled Compute Jobs
Virtual App Streaming
Integration
Simple Queuing
B2B Integration
Hybrid Connections
Pub/Sub Queuing
Media & CDN
Live & OD Media Streaming
Content Delivery Network (CDN)
Web and Mobile
Web Apps Infrastructure
API App Infrastructure
Mobile Backends
Business Process Automation
API Management
Push Notifications
Developer Services
Development Tools
Software Development Kits
Software Lifecycle Management
Application Instrumentation
Data
Relational SQL Database
Data Warehouse
Document Database Service
Distributed In-Memory Cache
Search
Simple Key/Value Store
Analytics & IoT
Big Data Analytics
Predictive Analytics
Data Stream Analytics
Big Data Storage
Data Pipelines
Device Data Collection
Data Source Management
IoT Device Management
Mobile Analytics
Hybrid Operations
Directory Health Monitoring
Privileged Identity Management
Domain Join & Policy Management
Server Data Backup
Operational Analytics
Bulk Data Import And Export
Disaster Recovery
Hybrid/Intelligent Data Backup
Infrastructure Services
OS/Server Compute
Virtual Servers
Containers
Storage
Disk-based Object/File Storage
Shared Storage
SSD-based Object/File Storage
Networking
Virtual Network
VM Load Balancer
DNS
Direct Network Connections
Traffic Distribution
VPN Gateway
HTTP Load Balancer
Datacenter Infrastructure (34 Regions, more than any other cloud provider)

https://www.youtube.com/watch?v=XfTRyF6TX6o

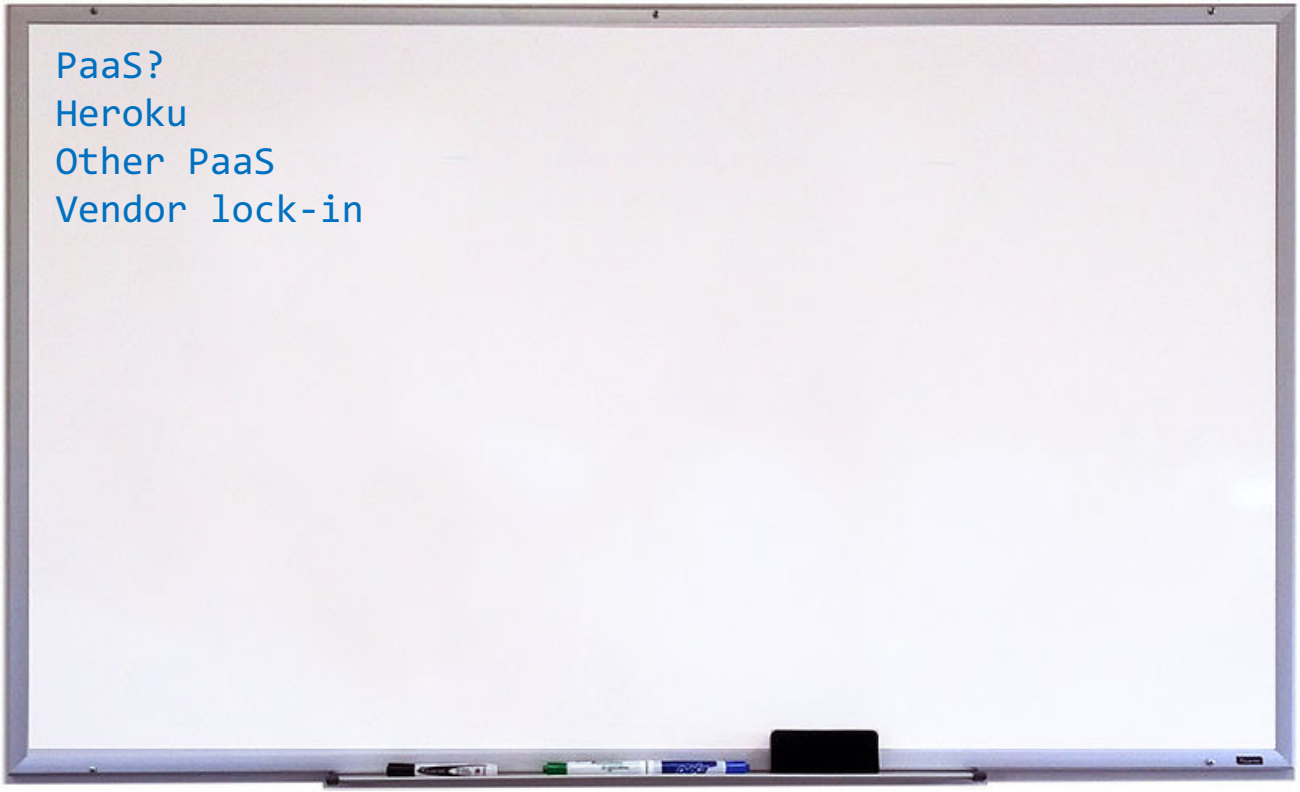
Firebase

➔lab

I would like to use a PaaS in production, with public hosting, autoscaling, supporting both Python and Go, and located in Europe.



https://paasfinder.org


PaaS?
Heroku
Other PaaS
Vendor lock-in

Definition - Vendor lock-in makes a customer dependent on a vendor for products or services, unable to use another vendor without substantial switching costs.

Example (PaaS)

- Federico* used GAE to develop a guestbook app featuring a "login" and a "write" button
- Federico wrote its app in Python using the Django framework, and he exploited Google Account API to implement user authentication

- Later on, Federico decided to change cloud provider to Windows Azure

- Azure featured an authentication framework allowing to access other services (including Google Account), but API had not been ported to Python yet $\rightarrow$ **authentication had to be reimplemented**
- To migrate users'files , he had to develop a **script** running on GAE that used both GAE and Azure libraries
- To migrate the DB: GAE's DB was not available as a single file that can be downloaded $\rightarrow$ **data had to be extracted brute-force** and stored in a simple format. The obtained file was then parsed by a **script** running on Azure.

**Federico Conte's BSc thesis, 2013*

Only 10% of organizations are "very concerned" about IaaS/PaaS public cloud vendor lock-in. Another 32% are "somewhat concerned”.

451 Research, 2020

Generally, the calculus in the enterprise market is shifting more toward speed, and we'll worry about platform later.

Dave Bartoletti, Forrester Research, 2020