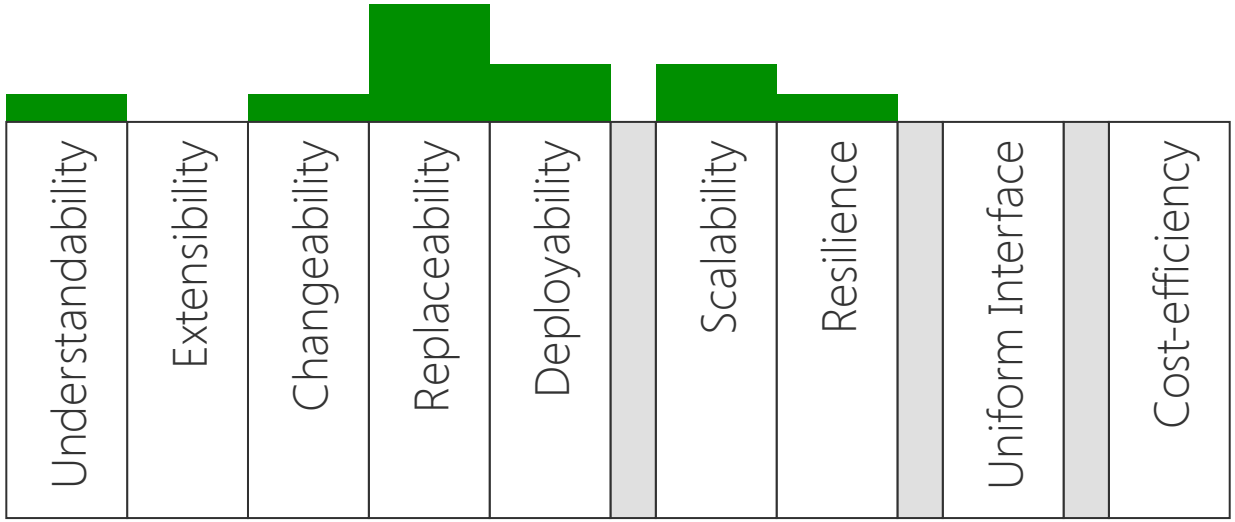
Let’s check a few hype topics …

µServices

- Built for replacement (not reuse)
- Self-dependent, loosely coupled services
- Should be aligned with business capability
- Size should not exceed what one brain can grasp

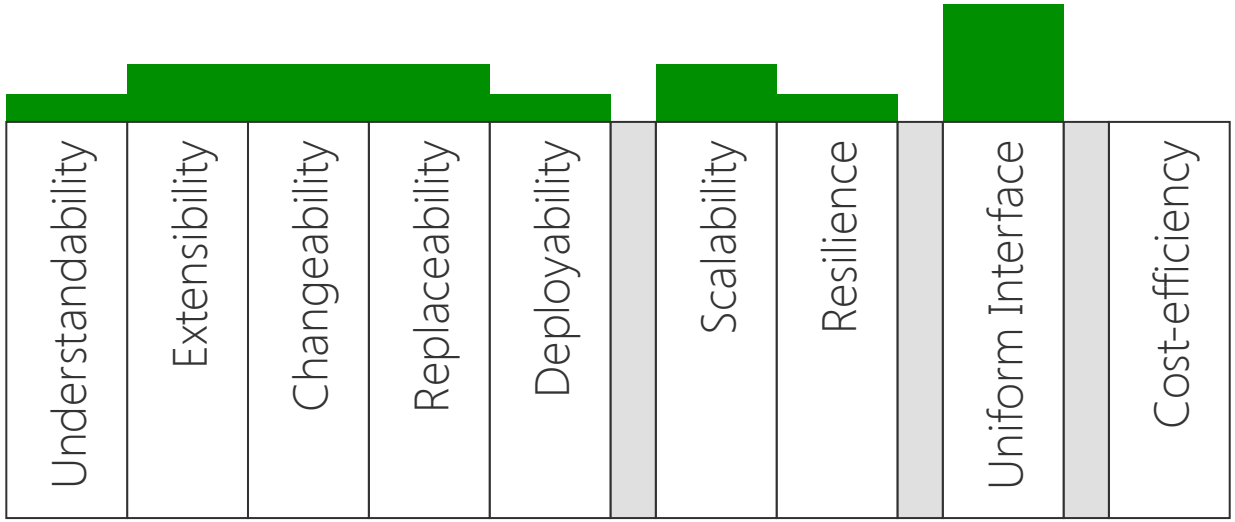
µServices


Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency

REST

- Uniform access interface to resources
- Closely related to the HTTP protocol
- HATEOAS (Hypermedia as the engine of application state)

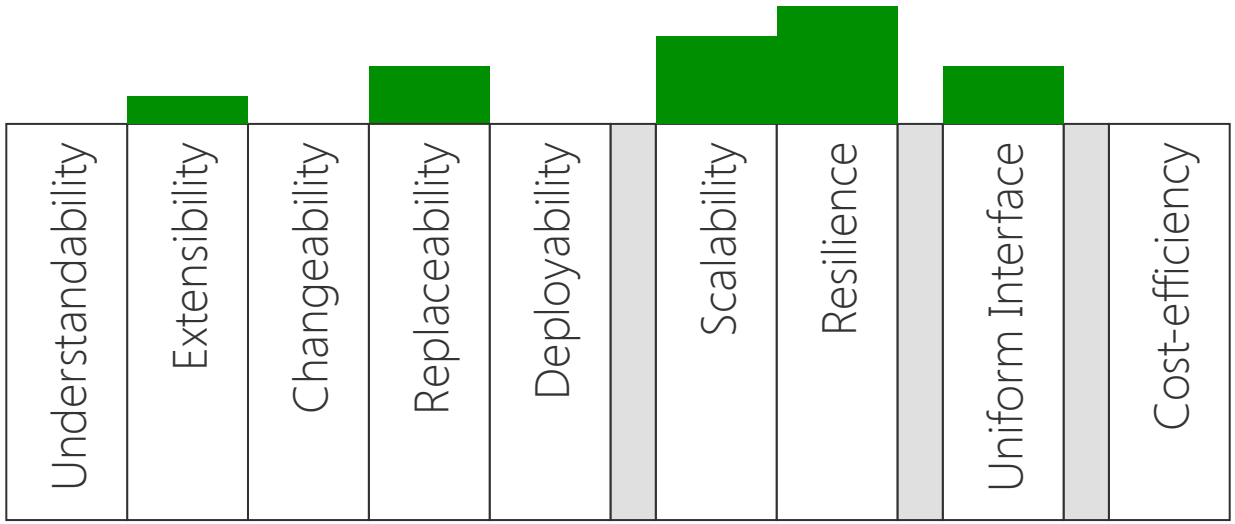
REST


Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency

Event/Message-driven

- Asynchronous communication paradigm
- Technical decoupling of communication peers (isolation)
- Location transparency in conjunction with MOM
- Call-stack paradigm replaced by (complex) message networks

Event/Message-driven

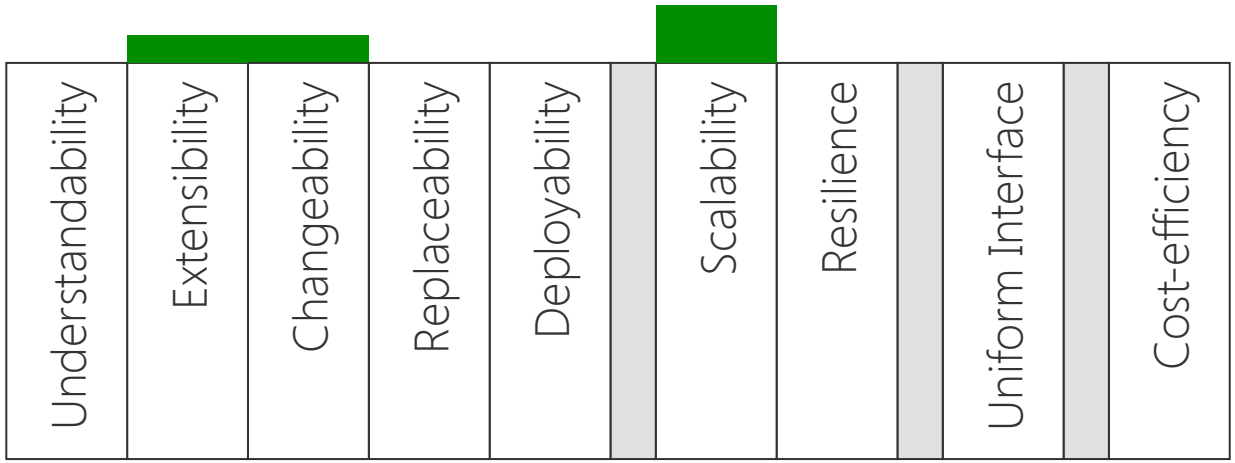


CQRS

- Command Query Responsibility Segregation
- Separate read and write interfaces including underlying models
- Separation can be extended up to the data store(s)
- Allows for optimized data representations and access logic

CQRS

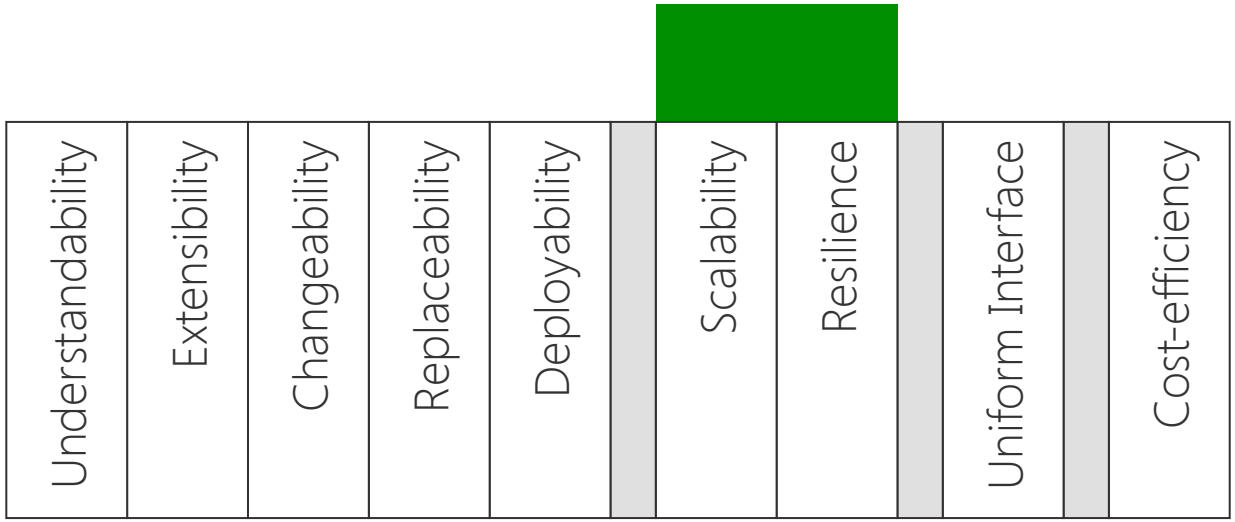

READ
WRITE


Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency

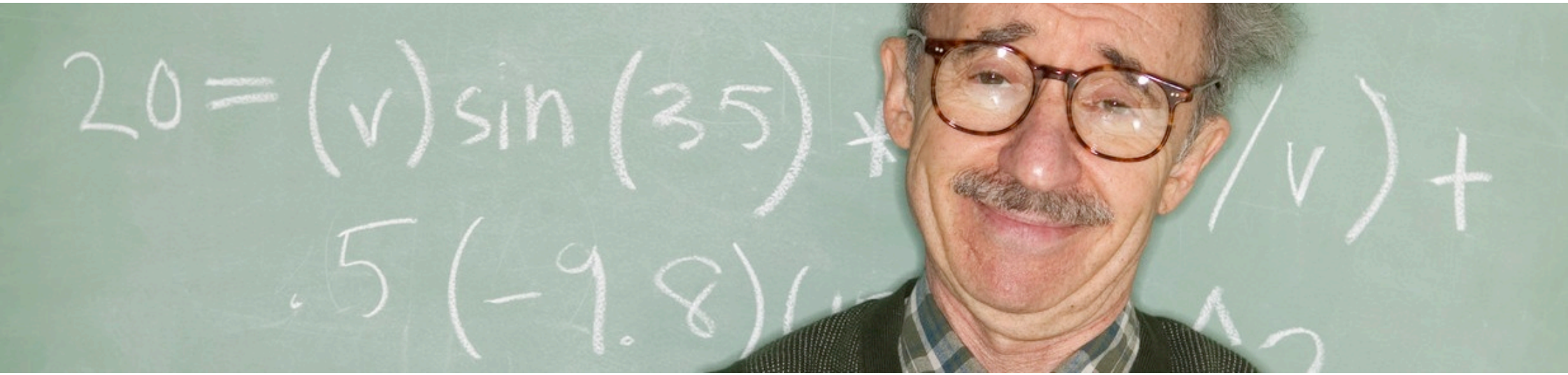
Reactive

- Message-driven – asynchronous and non-blocking
- Scalable – scaling out and embracing the network
- Resilient – isolation, loose coupling and hierarchical structure
- Responsive – latency control and graceful degradation of service

Reactive

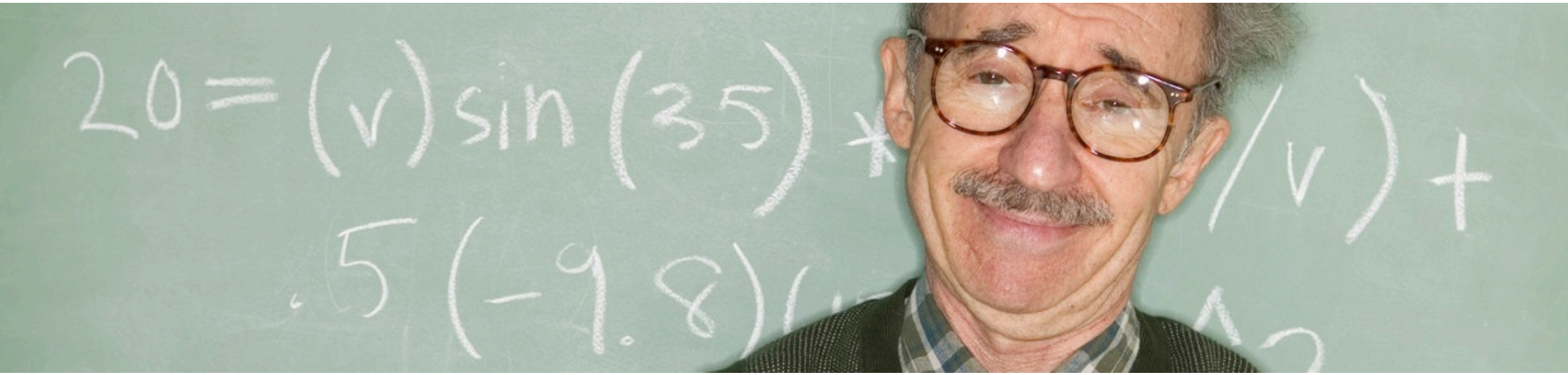

Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency

Functional Programming



- Alternative programming paradigm
- Functional languages (Erlang, Haskell, Clojure, …)
- Hybrid languages (Scala, …)
- Languages with functional extensions (Python, JavaScript, Java, …)

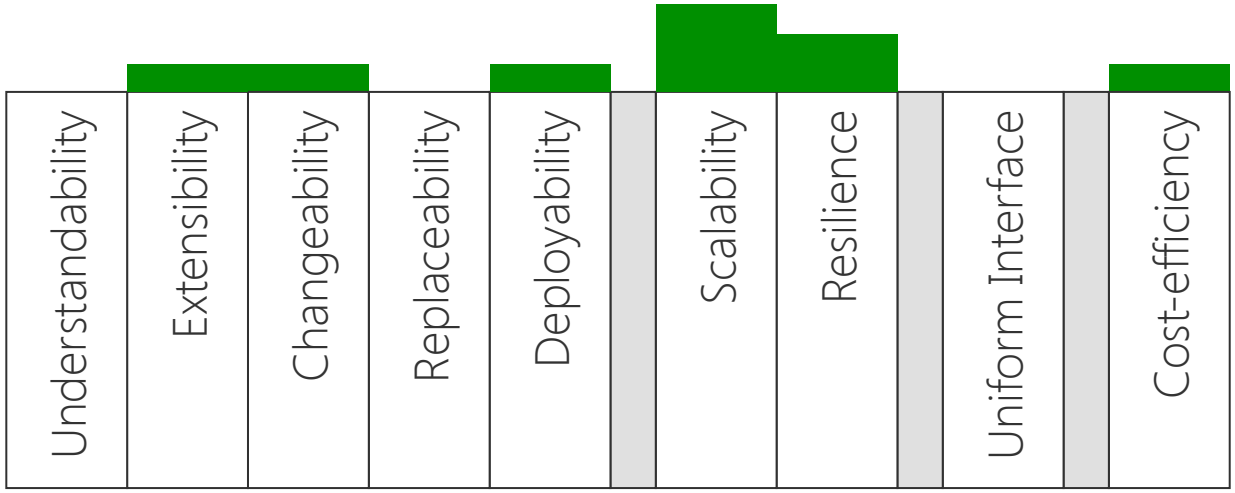
Functional Programming



NoSQL

- Augments the data store solution space
- Different sweet spots than RDBMS
- Key-Value Store – Wide Column Store – Document Store
- Graph Database

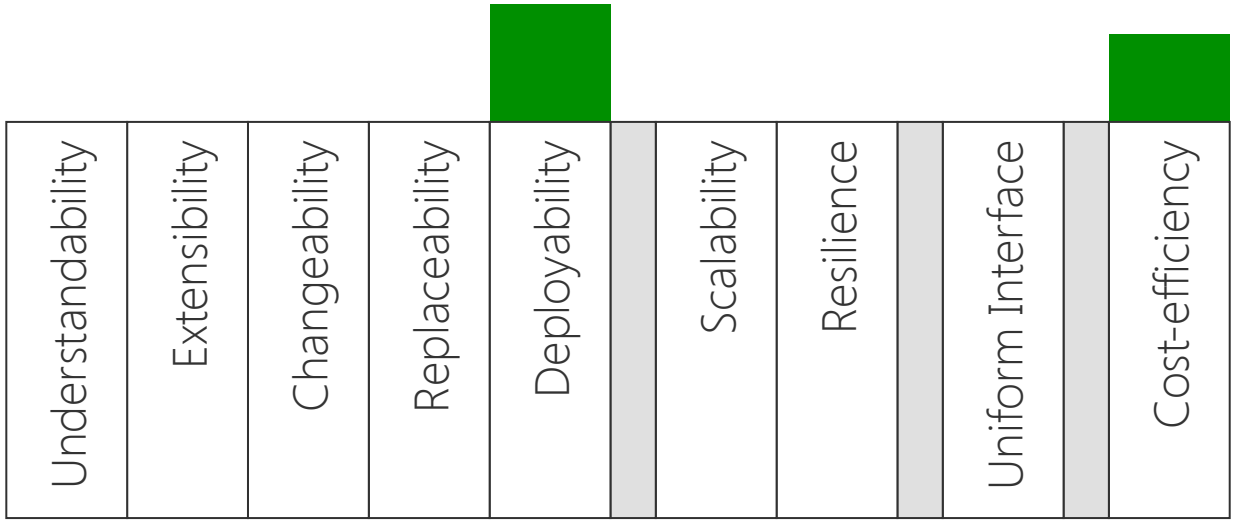
NoSQL


Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency

Continuous Delivery

- Automate the software delivery chain
- Build – Continuous Integration, …
- Test – Test Automation, …
- Deploy – Infrastructure as Code, …

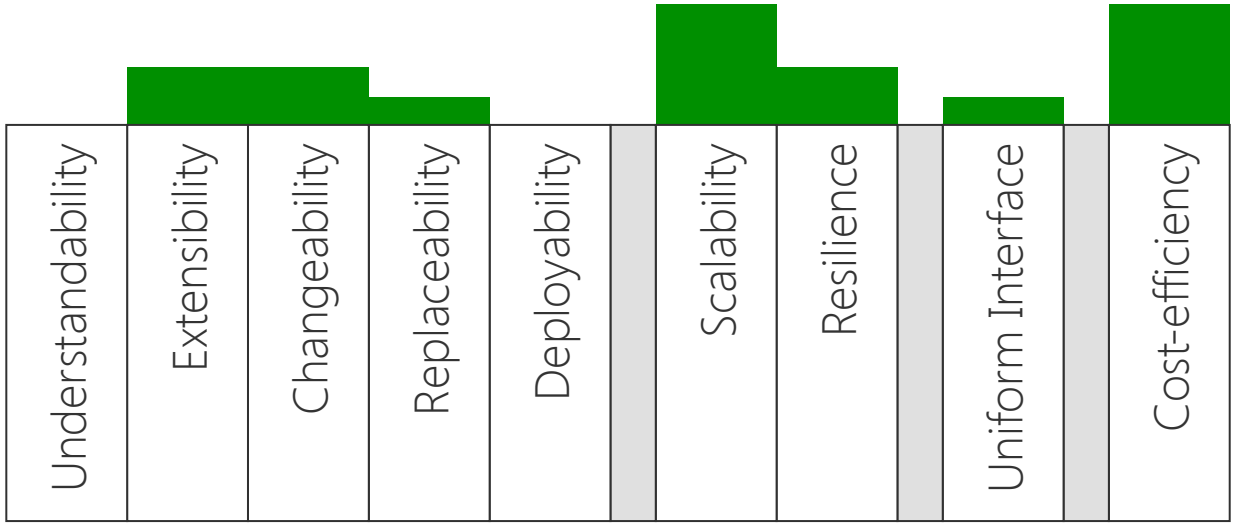
Continuous Delivery



Cloud provisioning model

- On-demand provisioning and de-provisioning
- Instant availability
- Self-service
- Pay-per-use

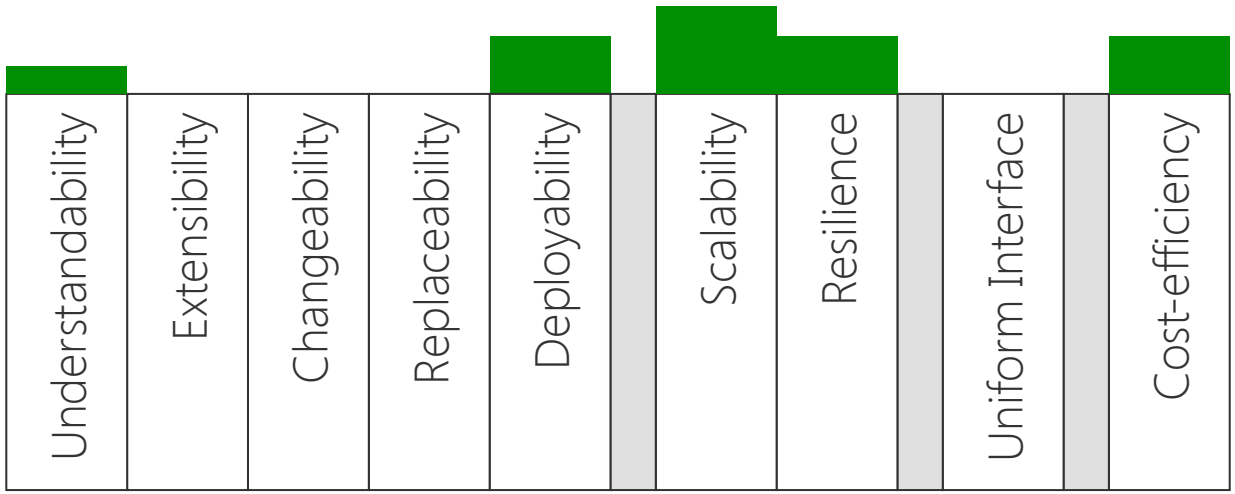
Cloud provisioning model



Docker

- Build, ship, run on container-basis
- Process-level isolation
- Declarative communication path configuration
- Cambrian explosion of ecosystem at the moment

Docker


Understandability
Extensibility
Changeability
Replaceability
Deployability
Scalability
Resilience
Uniform Interface
Cost-efficiency