

# Pico Services Architecture : Solving ALL problems for Microservices, REST and SOAP.

Authors: Adnan Masood, Mike Gold, David Ford

April 1st, 2016

## Abstract

The term "*Microservice Architecture*" has sprung up over the last few years to describe a 'particular' way of designing software applications. Like every new industry FAD<sup>1</sup>, including but not limited to Service Oriented Architecture (SOA), Microservices architecture has no precise definition, and it follows the "*it depends*" school of ivory tower software design. Following this prevalent and ubiquitous architectural style, we introduce a novel architectural design pattern called **(pico) p-Services Architecture**. Our architectural pattern addresses the prevalent characteristics around organizations such as maintaining and increasing technical debt, forming silos to decrease business capability, not leveraging automated deployment, ensuring lack of intelligence in the endpoints, and centralized<sup>2</sup> control of languages and data. Following are the tenets of the pico services architecture to help redirect focus away from the *minor* problems in enterprise distributed computing such as compliance, security, scalability, and fragmentation. In contrast with Micro-services architecture, Pico Services Architecture facilitates for the following three tenets of (fr)agile software design.

1. Following Conway's law[1], containers and hypervisors should be nested up-to nth level where n is the depth of organizational hierarchy for DevOps.

2. CAP (Consistency-Availability-Partition Tolerance) [2] theorem does not apply to p-services. Brewer's theorem after computer scientist Eric Brewer, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees. The p-services architecture prescribes to the monolith design and believes that ***if we work hard enough, we can achieve all three elements of CAP.***

3. The so called fallacies of distributed computing [3] are incorrectly believed to be false. In p-service architecture, every self-respecting software architect holds the following to be absolute, and unequivocally true statements.

- a. The network is *absolutely* reliable.
- b. Latency is *always* zero.

---

<sup>1</sup>Synonym for SOA, BPEL, WADL, etc

<sup>2</sup>Reviewer 2 thinks bottle neck sounds too pejorative.

- c. Bandwidth is *definitely* infinite<sup>3</sup>.
- d. The network is *perpetually* secure<sup>4</sup>.
- e. Topology doesn't change, *ever*.
- f. There is one administrator, *even* in the cloud.
- g. Transport cost is *always* zero.
- h. The network is *always* homogeneous

Following these three tenants, the **p-Services Architecture** provides a SOLID<sup>5</sup> foundation for enterprise architecture. We look forward to seeing pico Services in the IEEE annals of seminal Software references architectures.

## References:

- [1] Conway's Law of Software Design  
<https://www.thoughtworks.com/insights/blog/demystifying-conways-law>
- [2] CAP / Brewer's Theorem  
<https://groups.csail.mit.edu/tds/papers/Gilbert/Brewer2.pdf>
- [3] Fallacies of Distributed Computing  
[https://en.wikipedia.org/wiki/Fallacies\\_of\\_distributed\\_computing](https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing)

---

<sup>3</sup>Use Google Fiber

<sup>4</sup>Just ask NSA.

<sup>5</sup>No relation to SOLID OO pattern