

Why Reactive?

Operating in an Ever Changing World

Duncan K. DeVore

SWE @ Typesafe



What is Reactive?

First What Does it Mean?

What is Reactive?

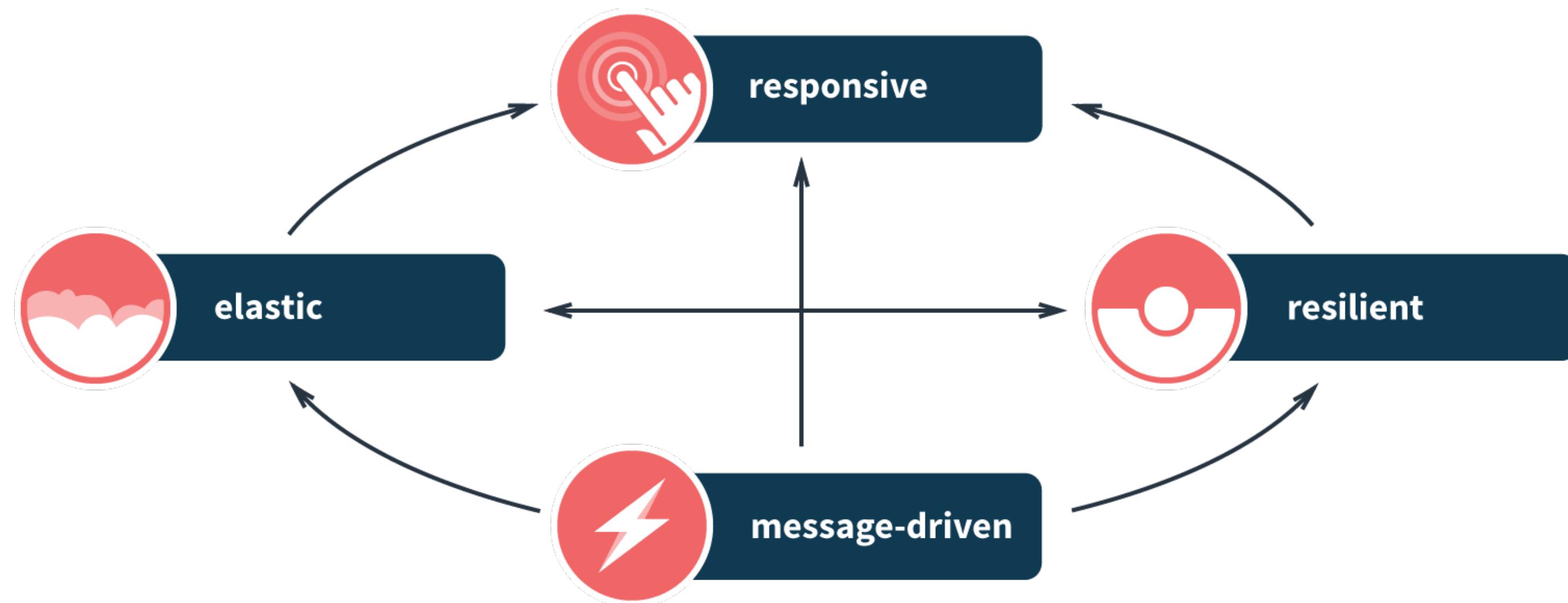
"Readily responsive to stimulus"

— Merriam Webster Dictionary

What is Reactive?

- Reactive Manifesto
- Published on September 16 2014. (v2.0)
- Jonas Bonér, Dave Farley, Roland Kuhn, and Martin Thompson
- 11K + Signatures

What is Reactive?



What is Reactive?

"*Responsive* - reacting in a desired or positive way; quick to react or respond"

— Merriam Webster Dictionary

What is Reactive?

"Elastic - able to return to an original shape or size after being stretched, squeezed, etc.; able to be changed"

— Merriam Webster Dictionary

What is Reactive?

"Resilient - able to become strong, healthy, or successful again after something bad happens"

— Merriam Webster Dictionary

What is Reactive?

"Message - a verbal, written, or recorded communication sent to or left for a recipient who cannot be contacted directly"

— Merriam Webster Dictionary

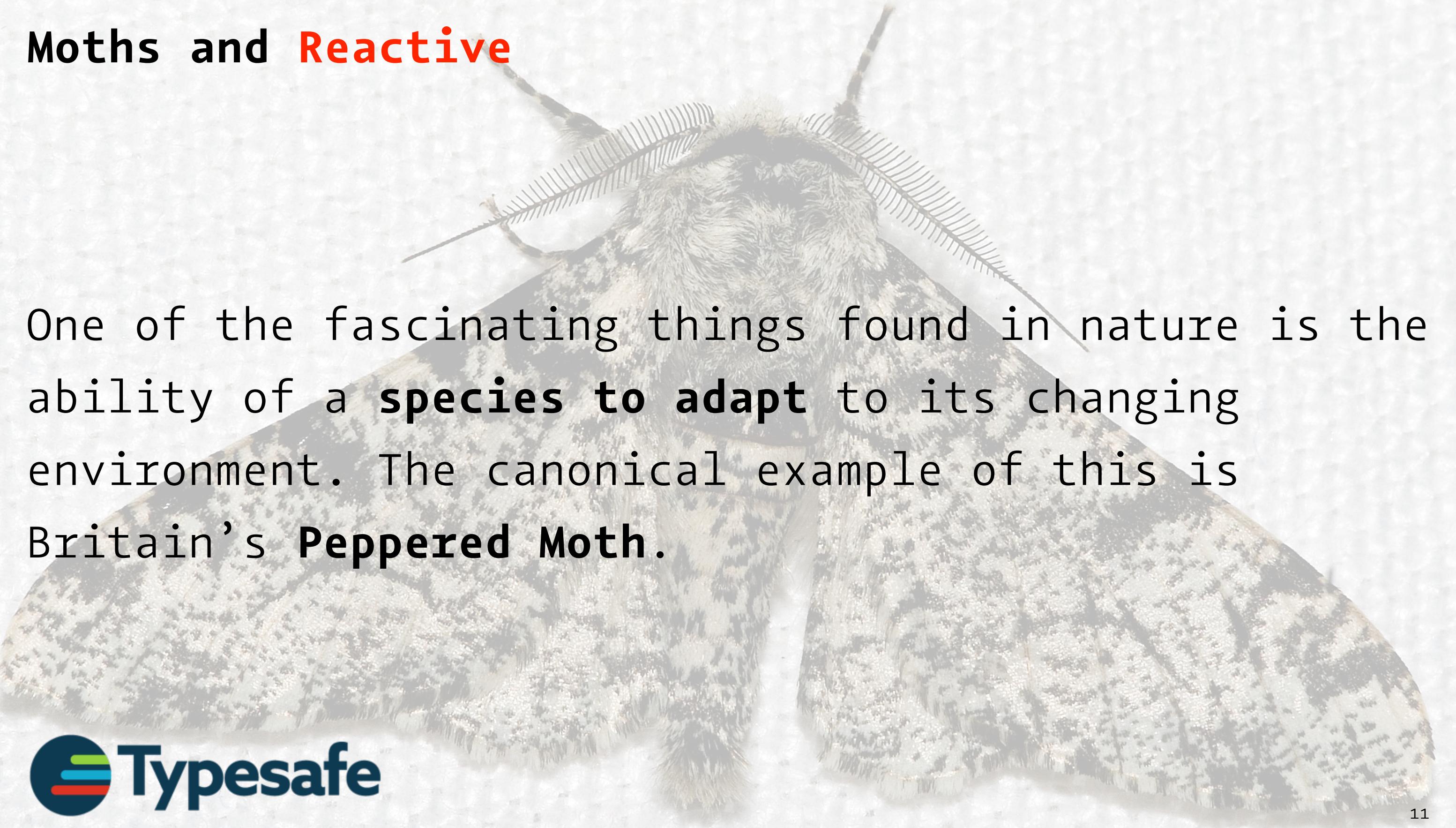
"Driven - operated, moved, or controlled by a specified person or source of power"

— Merriam Webster Dictionary

Moths and Reactive

An Example From Nature

Moths and Reactive



One of the fascinating things found in nature is the ability of a **species to adapt** to its changing environment. The canonical example of this is Britain's **Peppered Moth**.

Moths and Reactive

When newly **industrialized** Great Britain became polluted in the nineteenth century, slow-growing, **light-colored** lichens that covered trees **died** and resulted in a **blackening** of the trees bark.

The impact of this was quite **profound**!

Moths and Reactive

What are Lichens?

"Lichens - a simple slow-growing plant that typically forms a low crustlike, leaflike, or branching growth on rocks, walls, and trees."



Typesafe

Moths and Reactive

1. Light-colored moths, **camouflaged** and the **majority**
2. Found themselves the **obvious** target of hungry birds
3. Rare dark ones, **blended** into the polluted ecosystem
4. The birds, **changed** from eating dark to light moths
5. The dynamics of Britain's moth population **changed**

Moths and Reactive

Peppered Moth Before



Moths and Reactive

Peppered Moth After



Moths and **Reactive**

So what do **moths** have to do with reactive?

The peppered moth was able to **survive** due to a **mutation** that allowed it to **react** to its changing environment. This ability to react *on-the-fly* is what a **Reactive Application** does.

Moths and Reactive

How was the Peppered moth **Responsive**?

The Peppered moth **responded** to its environment by:

- Being **Elastic**
- Being **Resilient**
- Being **Message-Driven**

Moths and Reactive

How was the Peppered moth **Elastic**?

Load came in two forms:

1. **Unexpected**, increased pollution
2. **Resultant**, the lack of camouflage

The **mutation** baked into the Peppered moths DNA allowed the moth population to **rebound**.

Moths and Reactive

How was the Peppered moth **Resilient**?

Failure came in one form:

1. The **light-colored** moth's were easy targets

The **mutation** allowed the moth population to **respond** to this **failure**.

Moths and Reactive

How was the Peppered moth **Message-Driven?**

Organisms physiologic processes are **message-driven**¹.

- The **endocrine system** broadcasts its hormonal **messages** to essentially all cells by secretion into blood and extracellular fluid. Like a **radio broadcast**, it requires a **receiver** to get the message.

¹ The Body's Communication Systems

Moths and Reactive

How was the Peppered moth **Message-Driven**?

Organisms physiologic processes are **message-driven**¹.

- The **nervous system** exerts **point-to-point** control through **nerves**, similar to sending **messages** by conventional telephone. Nervous control is electrical in nature and fast.

¹ The Body's Communication Systems

Moths and Reactive

How was the Peppered moth Message-Driven?

Mutation the changing of the **structure of a gene**, resulting in a **variant** form that may be **transmitted** to subsequent generations, caused by the alteration of single base units in **DNA**, or the deletion, insertion, or rearrangement of larger sections of genes or chromosomes.

Moths and Reactive

How was the Peppered moth Message-Driven?

Multicellular organisms, **exchange DNA messages** only in the process of reproduction. This rule against multicellular genetic transfer is so universally obeyed that it is often referred to as "...the taboo of *intercellular transfer of genetic information.*"

Moths and Reactive

How was the Peppered moth **Message-Driven?**

Encoded in the **DNA** of the Peppered moth was a **message** that allowed the mutation to **respond!**

Business and Reactive

An Example From History

Business and **Reactive**

The Internet

- One of the greatest impacts in the last 50 years has been The **internet**
- The US commissioned research to build a **robust, fault-tolerant** computer network
- Began with a series of memos by **J.C.R. Licklider** of MIT in August 1962

Business and Reactive

The Internet

- Became known as the **Galactic Network** concept
- He envisioned a **globally** interconnected network of computers
- It would allow users to access data and programs from **anywhere** in the world

Business and Reactive

The Internet

- J.C.R. Licklider was the **Director** of the Information Processing Techniques Office
- (IPTO) was part of the Pentagon's **ARPA**, the Advanced Research Projects Agency
- Today, known as **DARPA**, the Defense Advanced Research Projects Agency

Business and Reactive Distributed Computing : Architecture

- A new computer model, **Distributed Systems** came into being
- It represented a **shift** in the computing paradigm
- Before, the model was **large, expensive** mainframe systems

Business and Reactive Distributed Computing : Architecture

- Affectionately referred to as **Big Iron**
- Mainframes used a **centralized** computing model
- Focusing on **efficiency, local scalability, and reliability**

Business and **Reactive**

Cloud Computing : Economics

- Distributed gave way to what we know as **Cloud Computing**
- A more powerful **less expensive** computing solution
- Cloud computing represents another **paradigm** shift

Business and **Reactive**

Cloud Computing : Economics

- Changing the way we **reason** about computer applications
- Distributed systems focus on the **technical** details
- Cloud computing focuses on the **economics** side of the equation

Business and **Reactive**

Storage, Network, CPU and Bandwidth

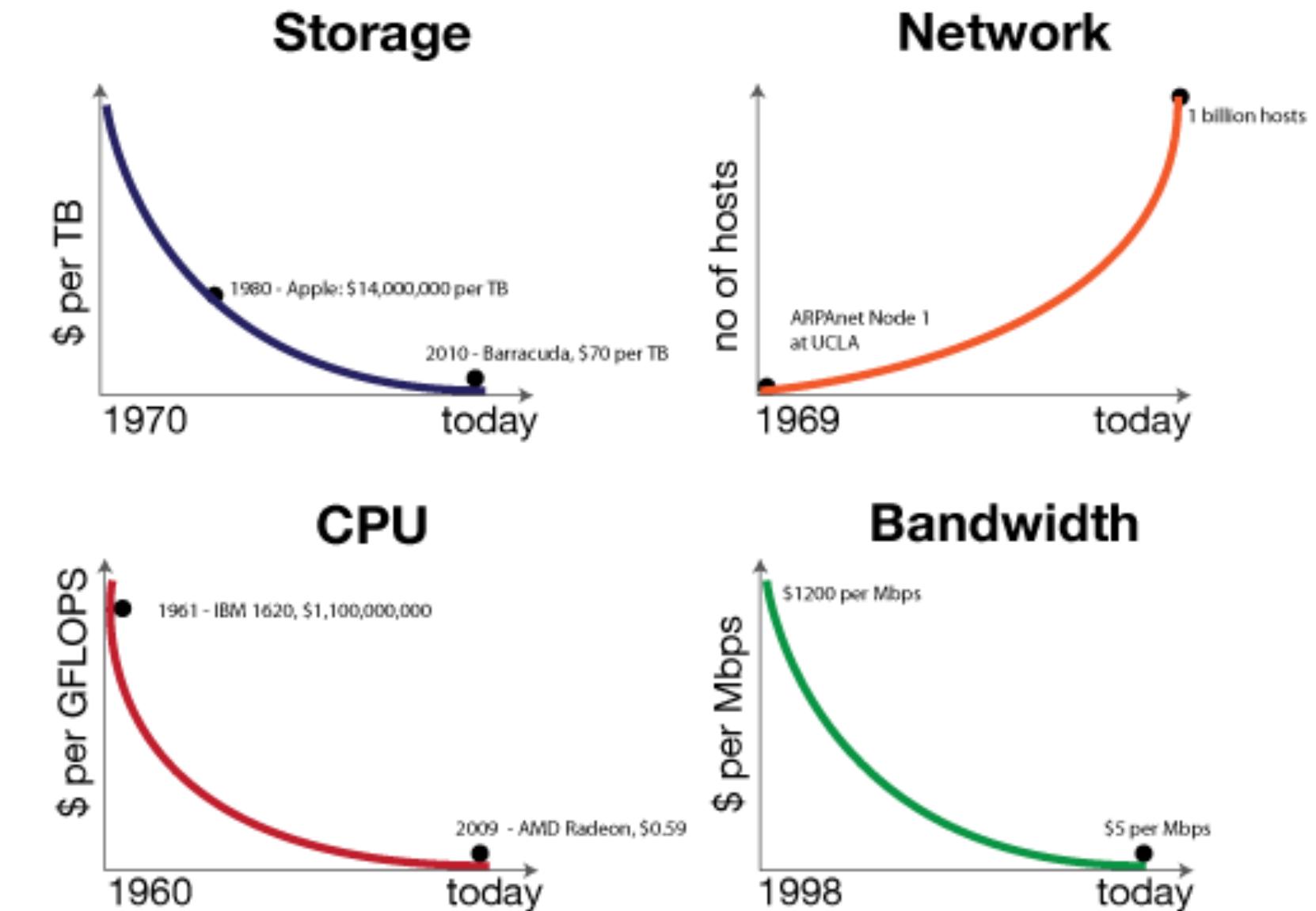
Recent hardware enhancement such as **multi-core** CPU's and **multi-socket** servers provide computing capabilities that were **non-existent** as little as 8 years ago.

Business and Reactive

Storage, Network, CPU and Bandwidth

The following shows the state of **storage**, **CPU**, and **bandwidth** compared to the number of network nodes. Notice the **increase** in use and the **decrease** in cost from the 70's!²

² image from [Oreilly Radar](#)

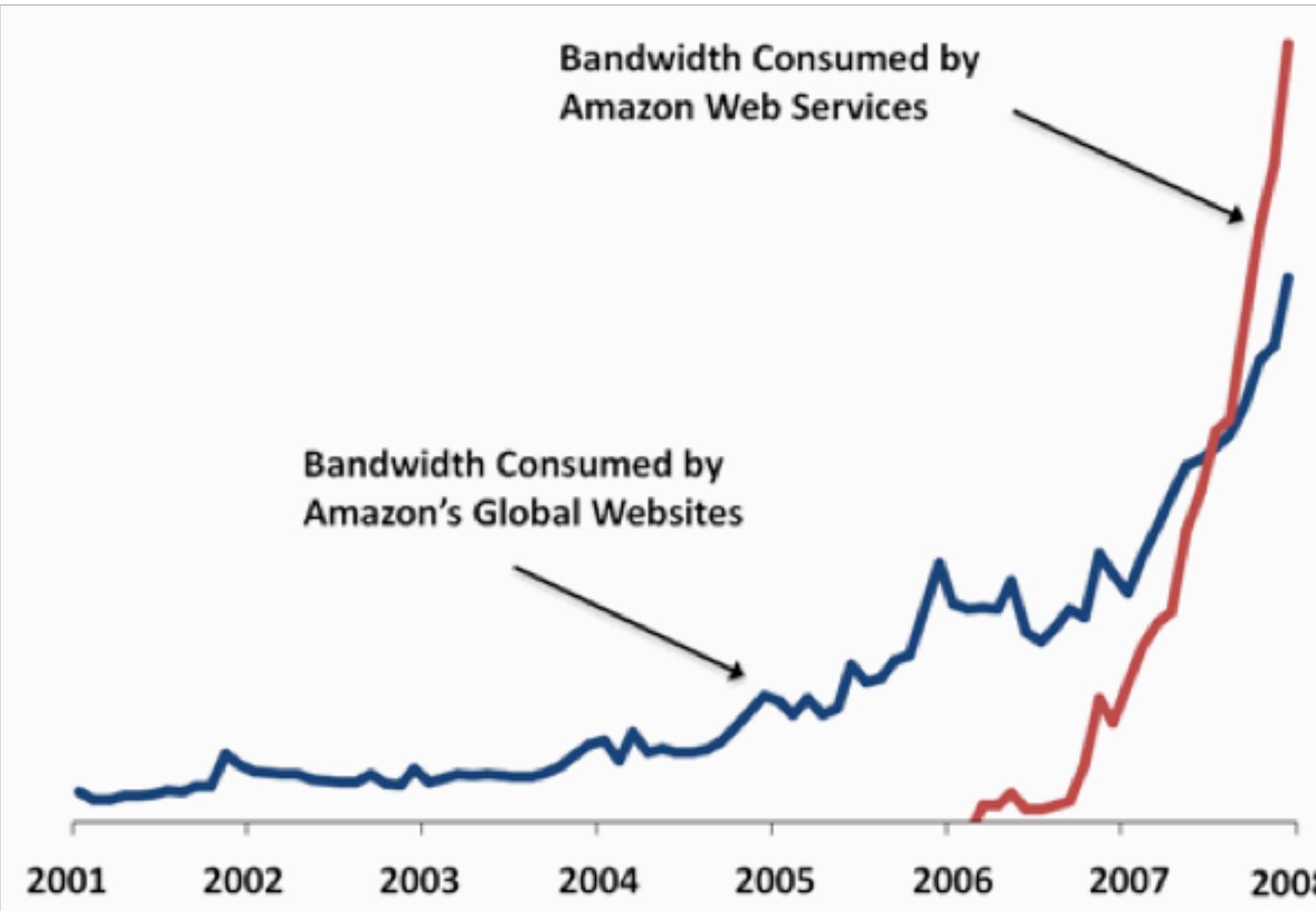


Business and **Reactive**

This new landscape of distributed cloud computing represents a **dramatic change** for the modern programmer, much like the Industrial Revolution of the nineteenth century did for the **Peppered** moth.

Business and **Reactive**

Rethinking One's Value Proposition



As a result, many companies have begun to **rethink** their **value proposition**:

- In January of 2008 Amazon announced that Amazon Web Services now **consume more bandwidth** than their entire global network of retail services, as shown in this figure from Amazon Blogs.³

³ image from [Amazon Blogs](#)

Business and Reactive

Rethinking One's Value Proposition

What is Amazon? An **online bookstore or provider of Cloud Services?**

Business and Reactive

"As user expectations of sub-second performance, spikes in application load, demands to run on multi-core hardware for parallelism, and data needs expand into the petabytes...,"

Business and Reactive

...modern applications must embrace these changes by incorporating reactive behavior into their DNA."

— Reactive Application Development

Programmers and Reactive

Details, Details, Details ...

Programmers and Reactive

Message-Driven : Key Ideas

- Based on **asynchronous** communication
- Sender/recipient **not affected** by **message propagation**
- You can design your system in **isolation**
- No **worries** how the transmission of messages occurs
- **Message-driven** communication => a **loosely** coupled design
- Provides **scalability**, **resilience** and **responsiveness**

Message-Driven Don't's

NEVER



NEVER
EVER



NEVER
EVER
BLOCK



Message-Driven Do's

Go Async

Programmers and Reactive

Message-Driven : Go Async

- Actors
- Conflict Free Resolution Datasets (CRDTs)
- Reactive Streams
- Futures

Programmers and **Reactive**

Elasticity : Key Ideas

- System stays **responsive** under varying workload
- Actively scale **up/down**, **in/out** based upon usage
- Saves **money** on unused computing power
- Ensures the **servicing** of **growth** in user base

Elasticity Don't's

**NEVER
EVER
BLOCK**



Programmers and Reactive

Elasticity : Never Ever Block

Universal Scalability Law

It's been proven that **blocking** of any kind, anywhere in the system will **measurably** impact scale due to:

1. **Contention**; waiting for queues or shared resources.
2. **Coherency**; the delay for data to become consistent

Elasticity Do's

Programmers and **Reactive**

Elasticity : Distributed By Default

Location Transparency

1. **Explicit** distributed computing
2. **Local** communication is an optimization
3. **Locality** of data
4. **Async** message passing

Programmers and **Reactive**

Elasticity : Share Nothing

If multiple threads access the same mutable state variable without synchronization:

- Your program is broken.



Programmers and **Reactive**

Elasticity : Share Nothing

- Embrace **Immutability**
- Immutable collections
- Defensive copies
- Mutability in **local thread-safe** state **only**



Programmers and **Reactive**

Resilience : Key Ideas

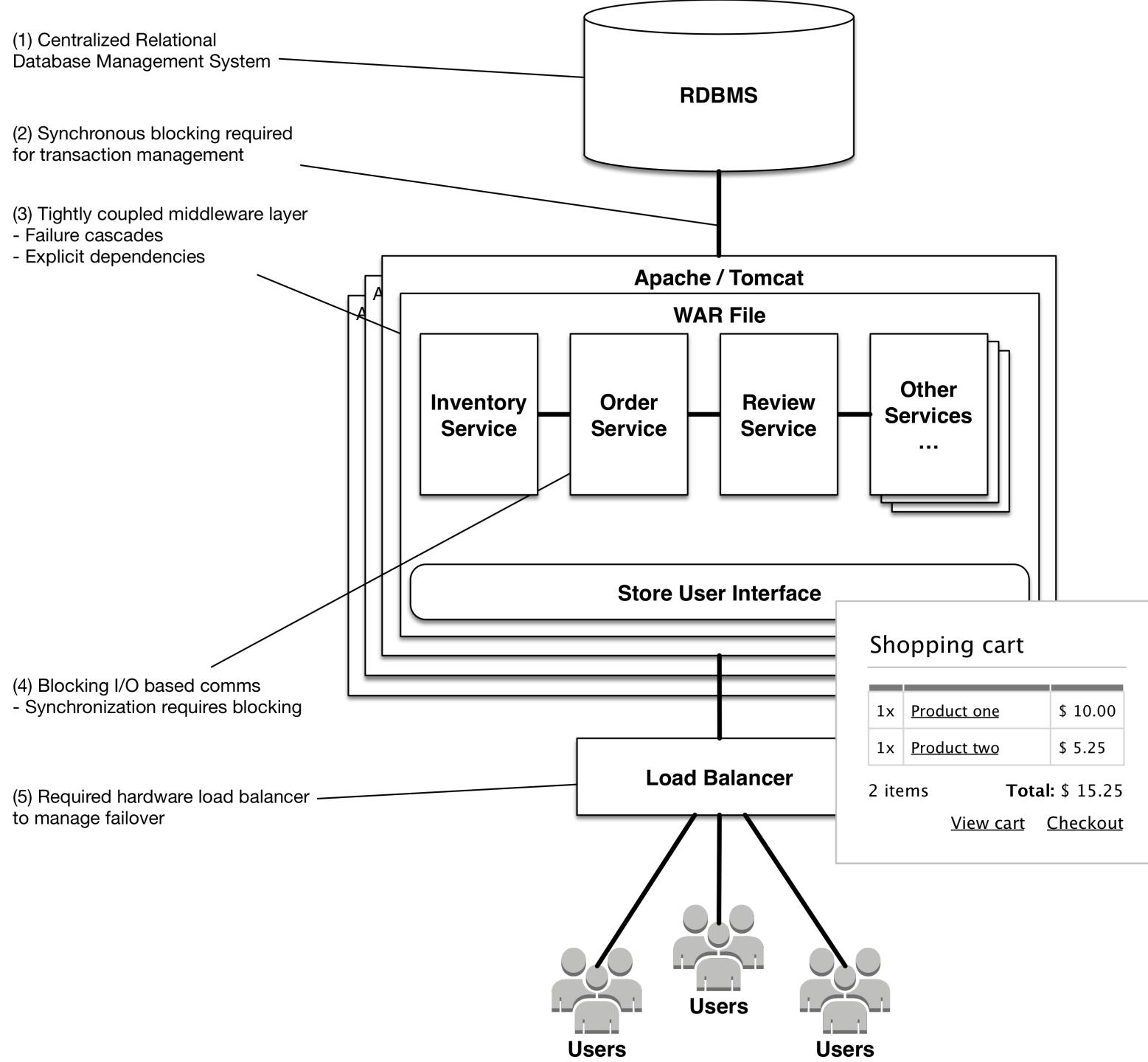
- System stays responsive in the **face of failure**
- Failure is **expected/embraced**; systems exist in **isolation**
- A **single** point of failure **remains** just that
- The system **responds** appropriately
- Strategies for **restarting** or **re-provisioning**
- **Seamless** to the overall systems

Resilience Don't's

Programmers and Reactive

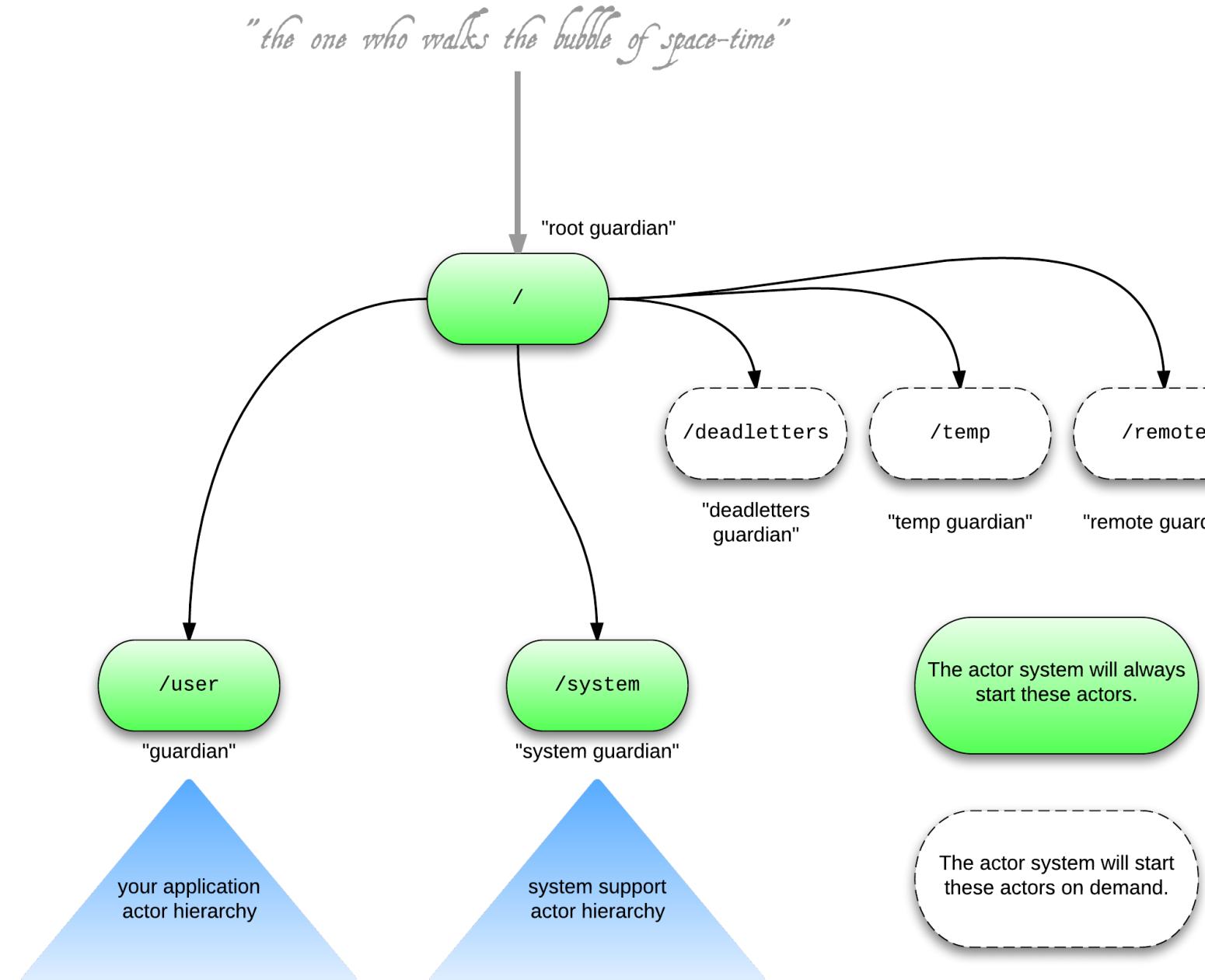
Resilience : No Monoliths

1. Centralized RDBMS
2. Synchronous blocking
3. Tightly coupled middleware
4. Blocking I/O based comms
5. Required load balancer



Resilience Do's

Programmers and Reactive Resilience : Supervision



1. **Resume** the subordinate
2. **Restart** the subordinate
3. **Stop** the subordinate
4. **Escalate** the failure

Programmers and Reactive

Resilience : Use Bulkheads

1. Isolate failure
2. Compartmentalize
3. Manage failure locally
4. Avoid cascading failures



Programmers and **Reactive**

Responsiveness : Key Ideas

- **Cornerstone** of usability and utility
- The system **responds** promptly if at all possible
- Means that **problems** may be **detected** quickly
- Means that **problems** are **dealt** with effectively

Programmers and Reactive

Responsiveness : Everyone is Happy

- Customers stay **happy!**
- Bosses stay **happy!!**
- You **happy!!!**

Results of Non-Reactive

Results of Non-Reactive?

"In the new world, it is not the big fish which eats the small fish, it's the fast fish which eats the slow fish."

— Klaus Schwab

Results of Non-Reactive?

"Everyone has a plan 'till they get punched in the mouth."

— Mike Tyson



The End