Event-Driven Architecture Workshop

THREE AMIGO'S HEALTH

Pre-Work

- Ensure you have an Azure Subscription
  - http://portal.azure.com
  - Could be around $5 in charges to complete workshop, but there’s a $200 when you first signup and you may be eligible for an ongoing monthly Visual Studio credit.
- Ensure you have Visual Studio installed
  - Community version is fine
- Clone https://github.com/scottctr/EdaWorkshop
- Introduction to event-driven architecture article
  - https://developer.ibm.com/articles/advantages-of-an-event-driven-architecture/

Overview

- What are we going to learn?
- How are we going to learn it?
- How long until we start writing some code?

Who Are We?

- Developers using event-driven architecture
- Impressed with improvements over other architectures and want to help spread the word
- Worked with a few NSS graduates and glad to be a part of that wonderful organization as well

What are we going to learn?

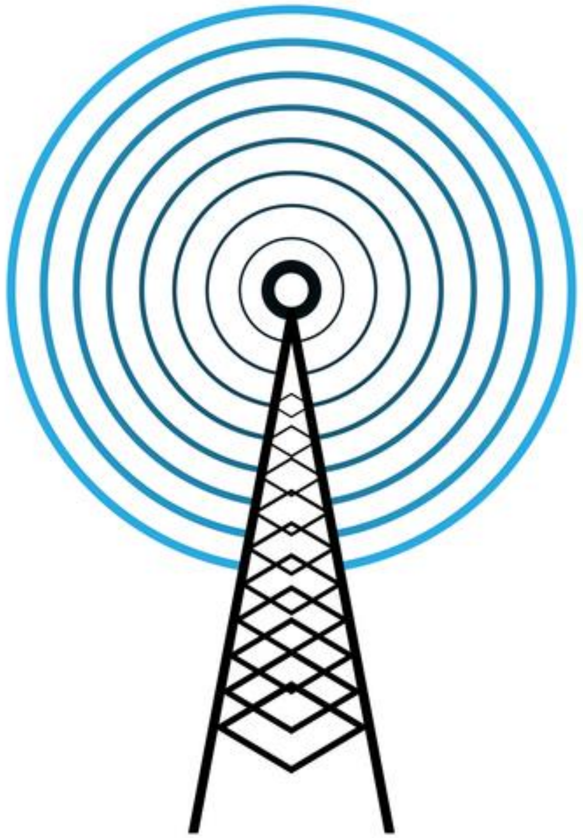
- Event-driven architecture (EDA) design concepts
- How to use EDA to improve your solutions
- How to implement an EDA solution on Azure

How are we going to learn?

- Start by looking at issues with traditional systems
- Define goals for an improved system
- Walk through EDA solution design
  - Introduce key terms and concepts
- Implement EDA solution on Azure
  - More key terms and concepts
  - Follow an agile/scrum structure

Overly Simplified Analogy

VS

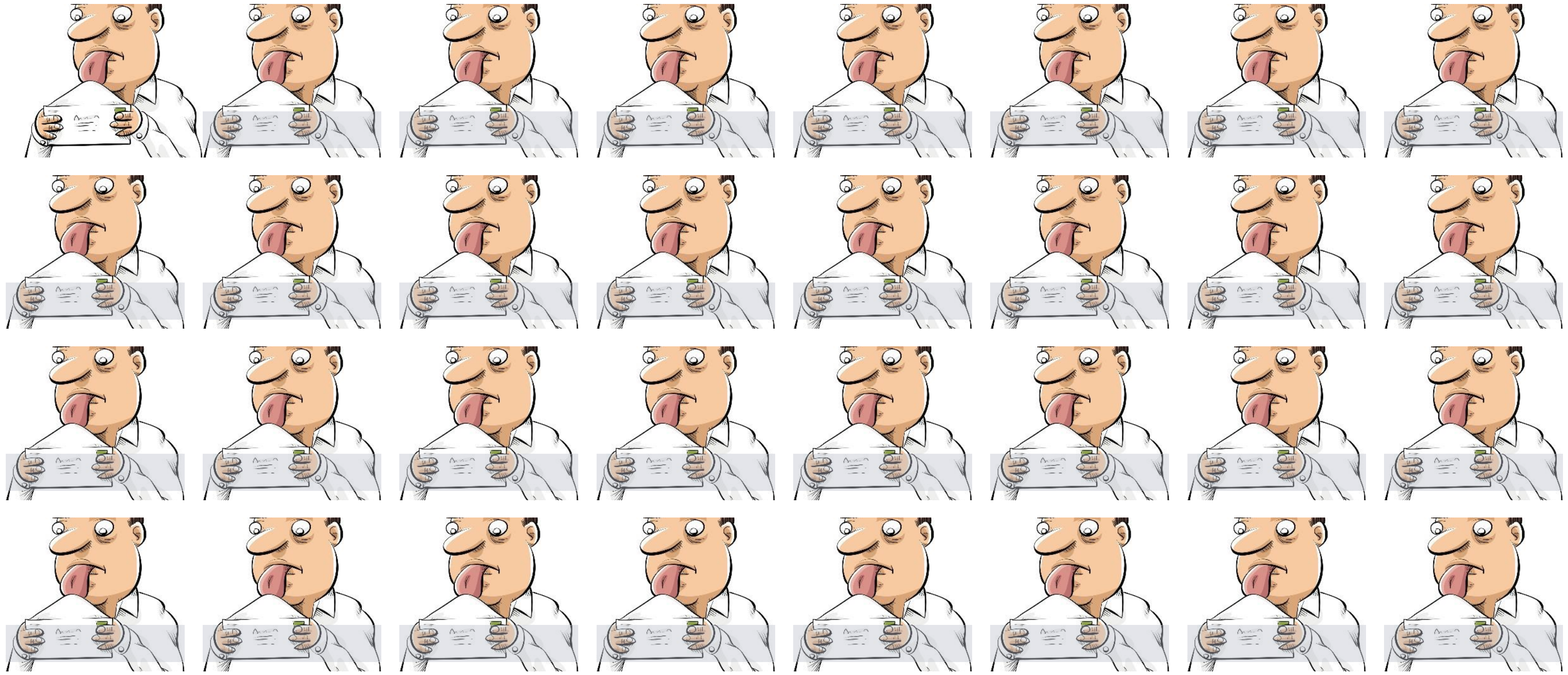


Overly Simplified Analogy

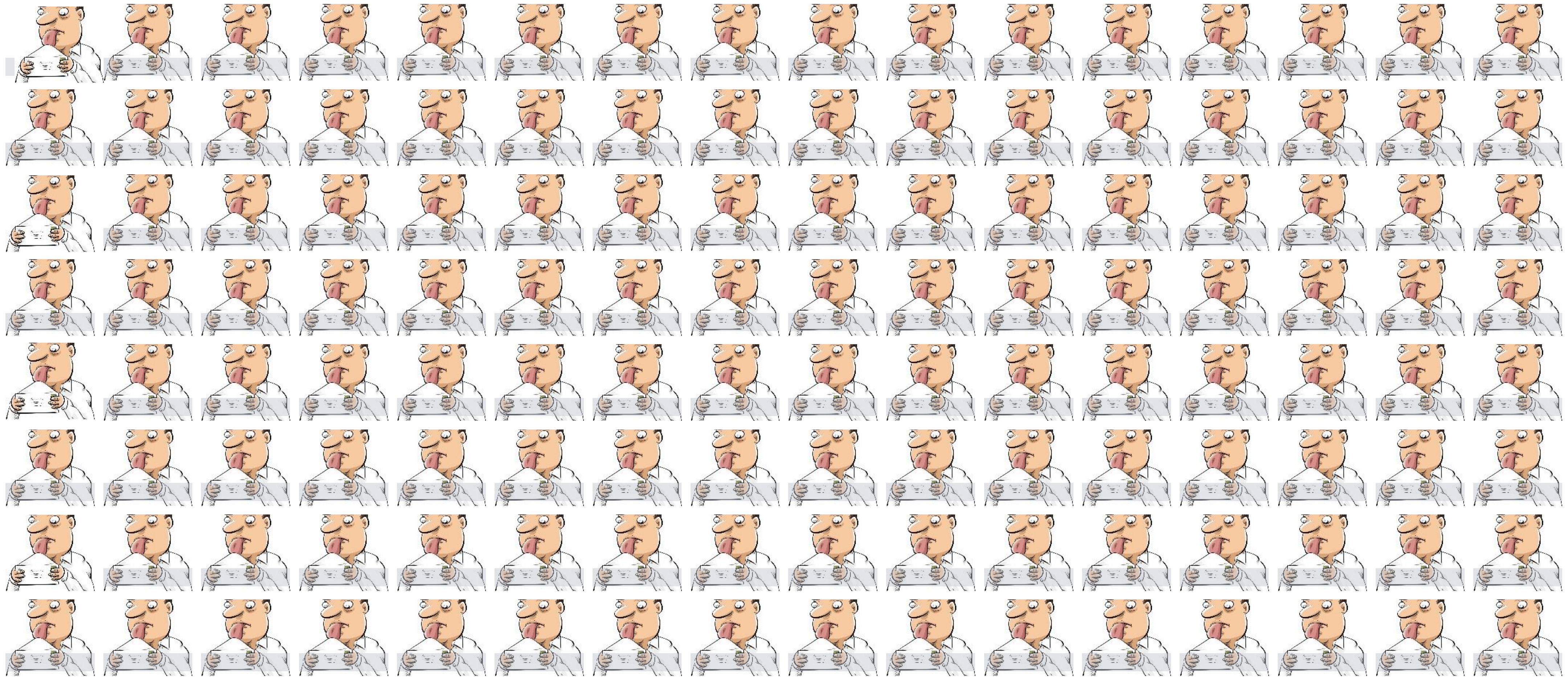
Overly Simplified Analogy

Overly Simplified Analogy

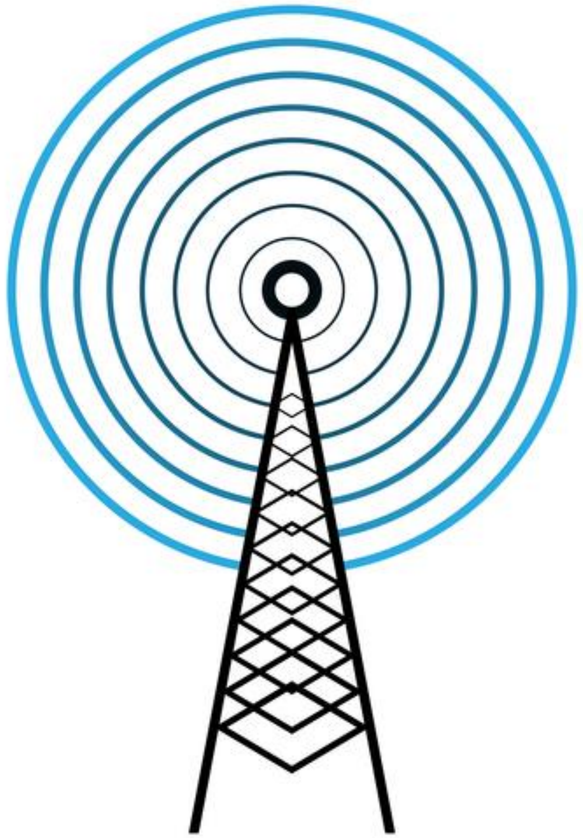
Overly Simplified Analogy



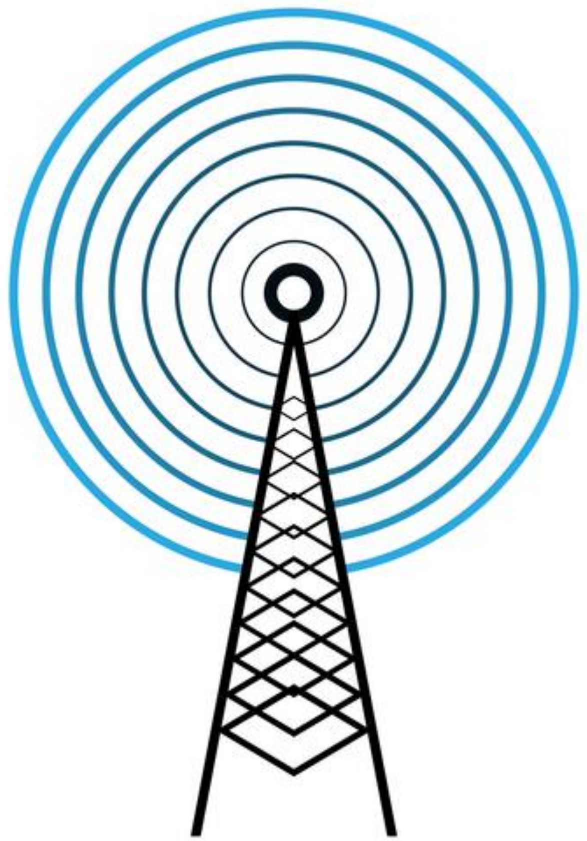
Overly Simplified Analogy



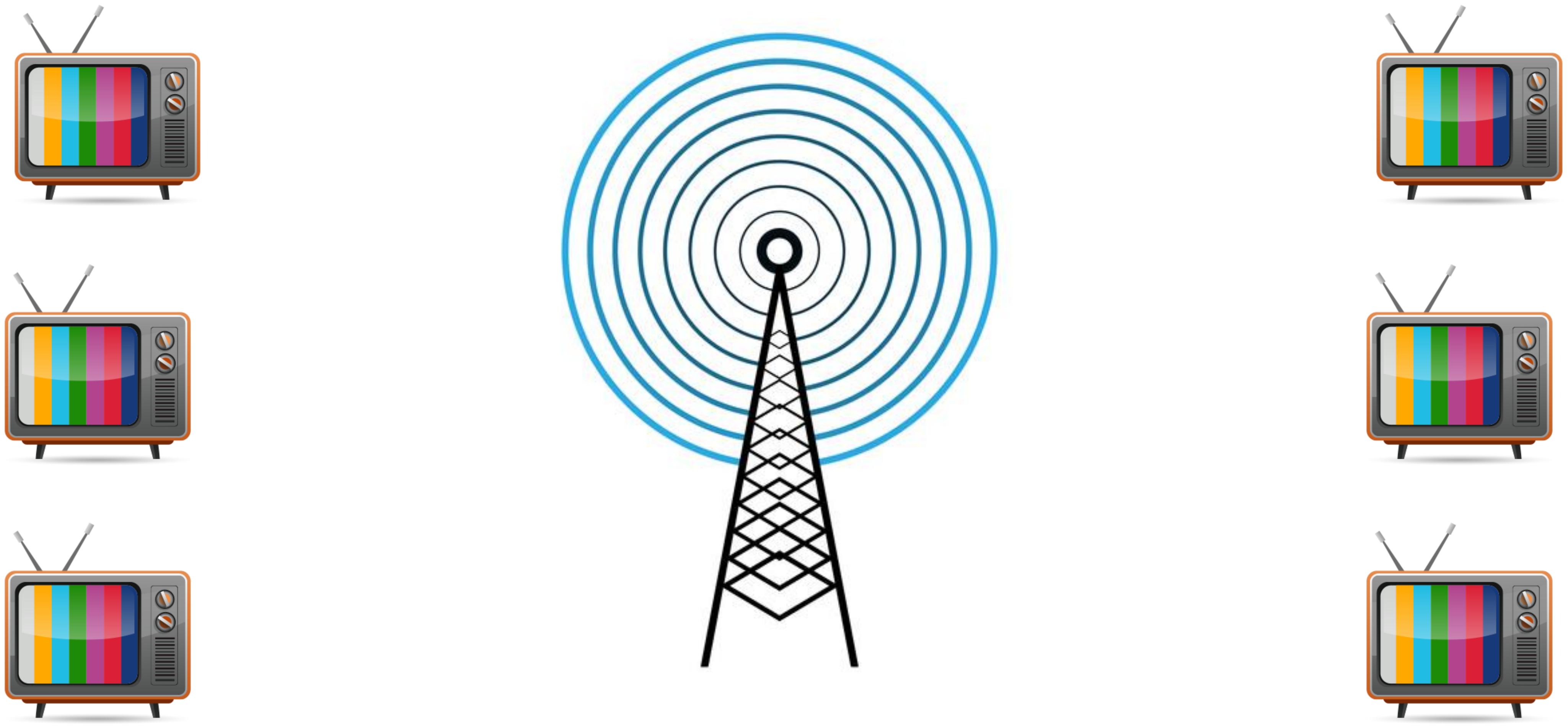
Overly Simplified Analogy



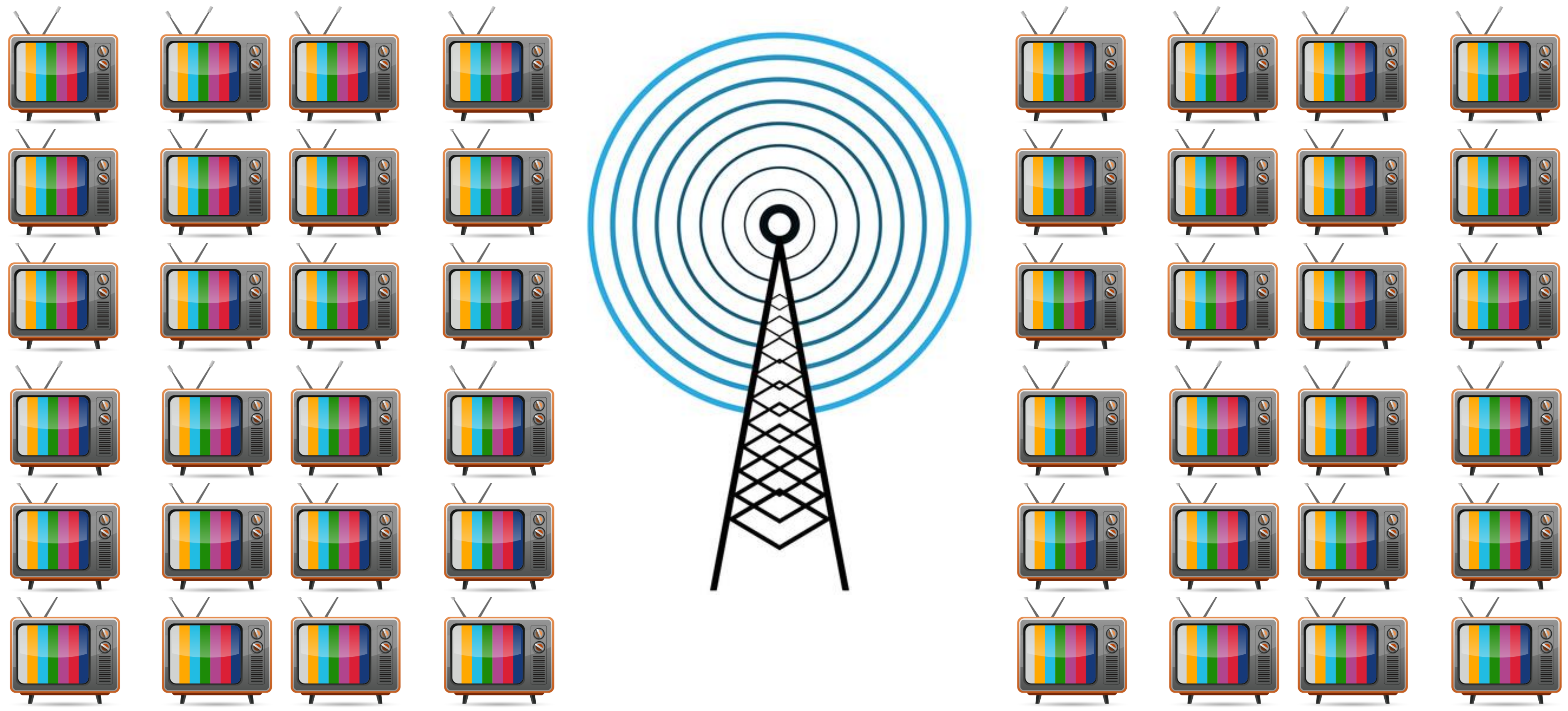
Overly Simplified Analogy



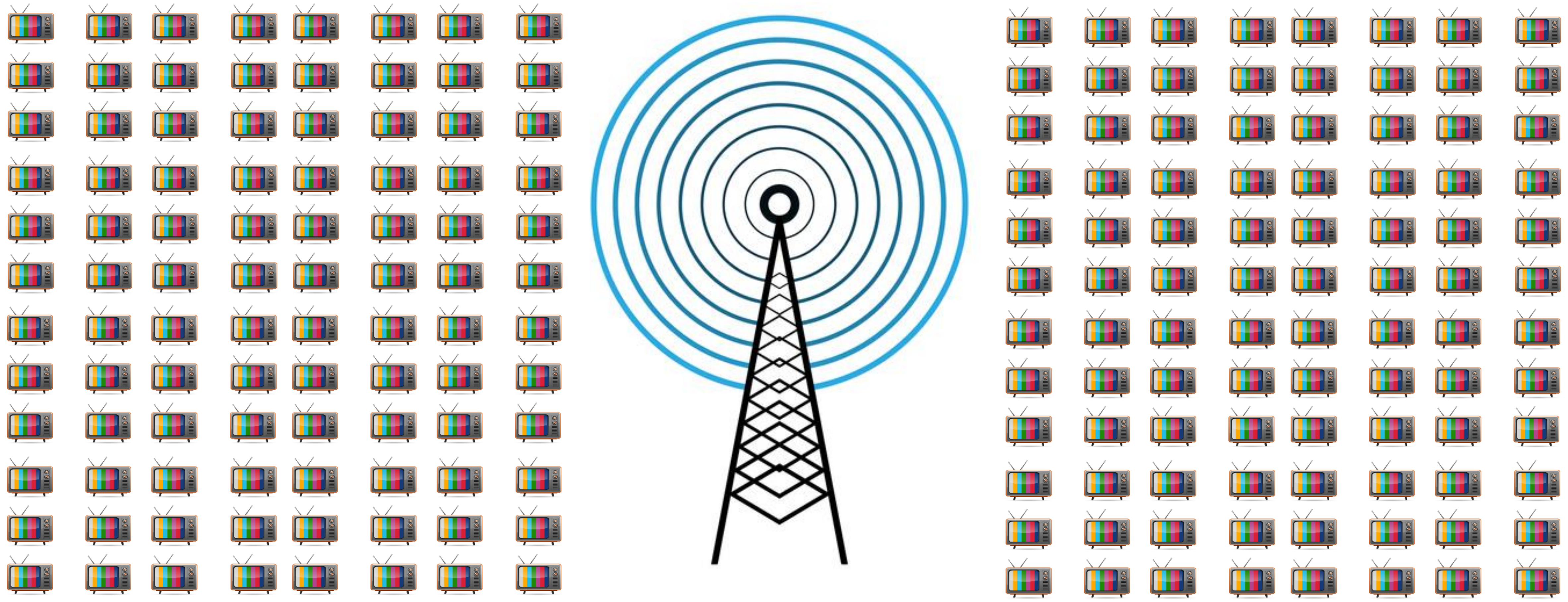
Overly Simplified Analogy



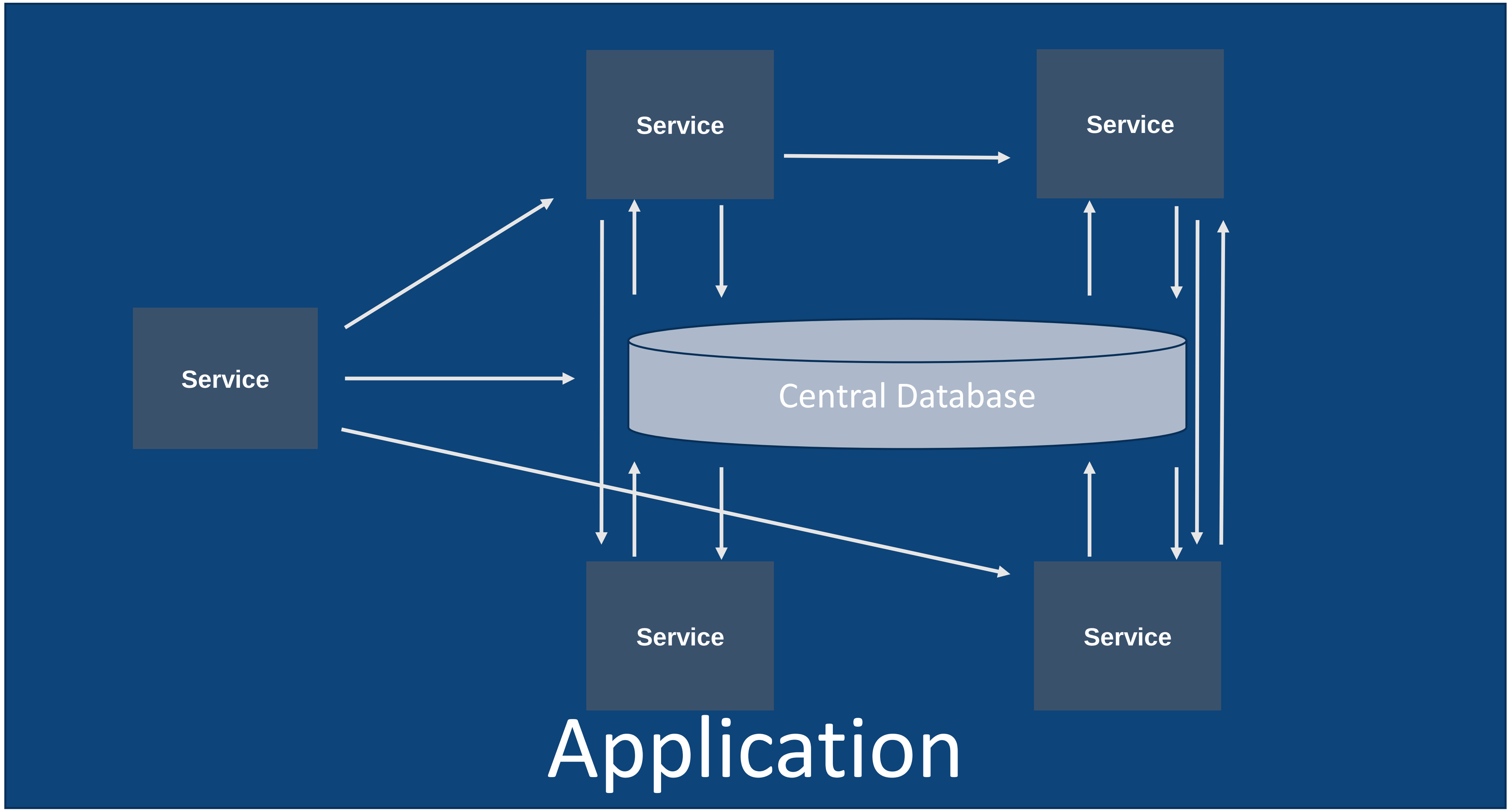
Overly Simplified Analogy



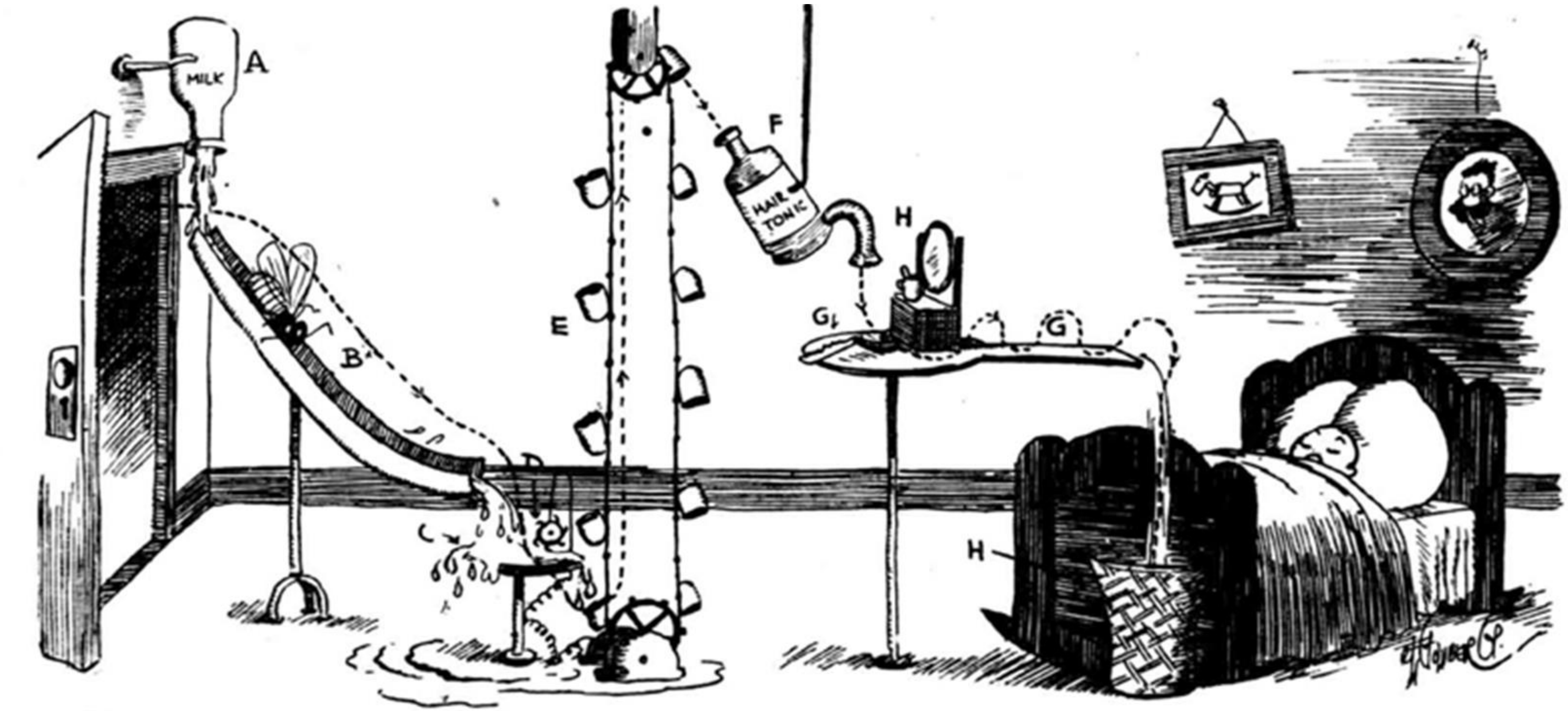
Overly Simplified Analogy



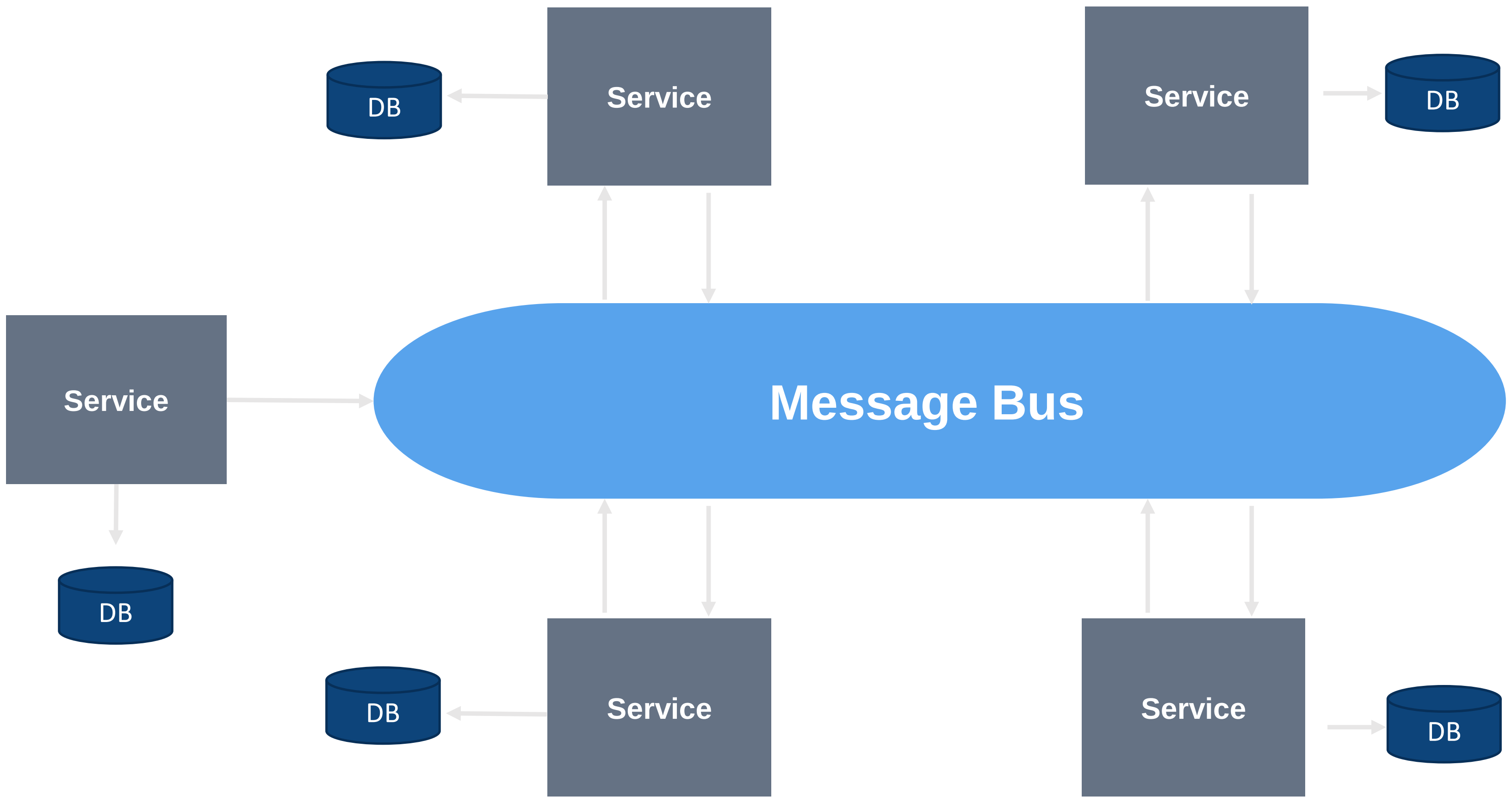
Old School Architecture



Highly Coupled Systems



EDA Architecture



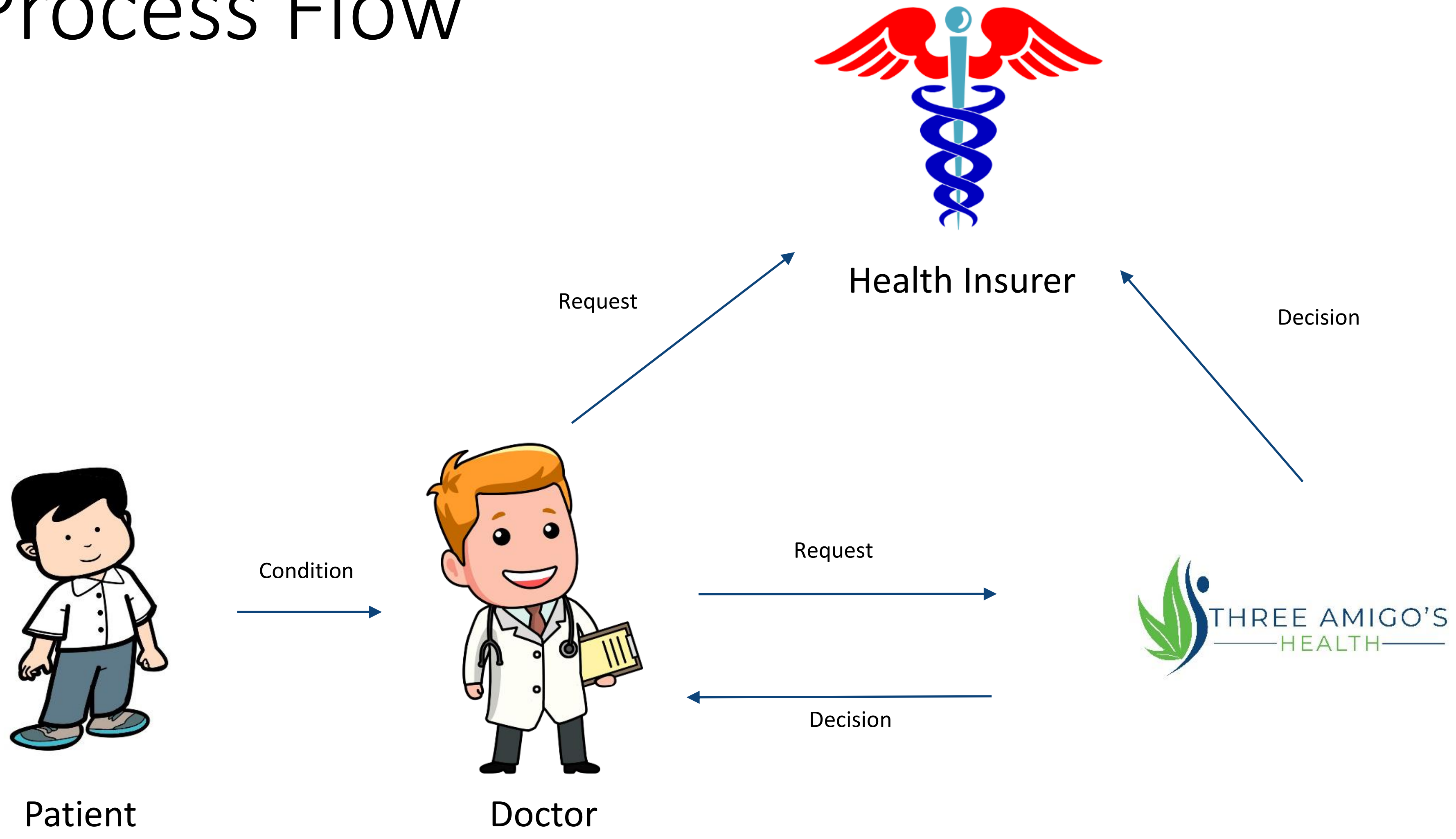
Context

THREE AMIGO'S HEALTH

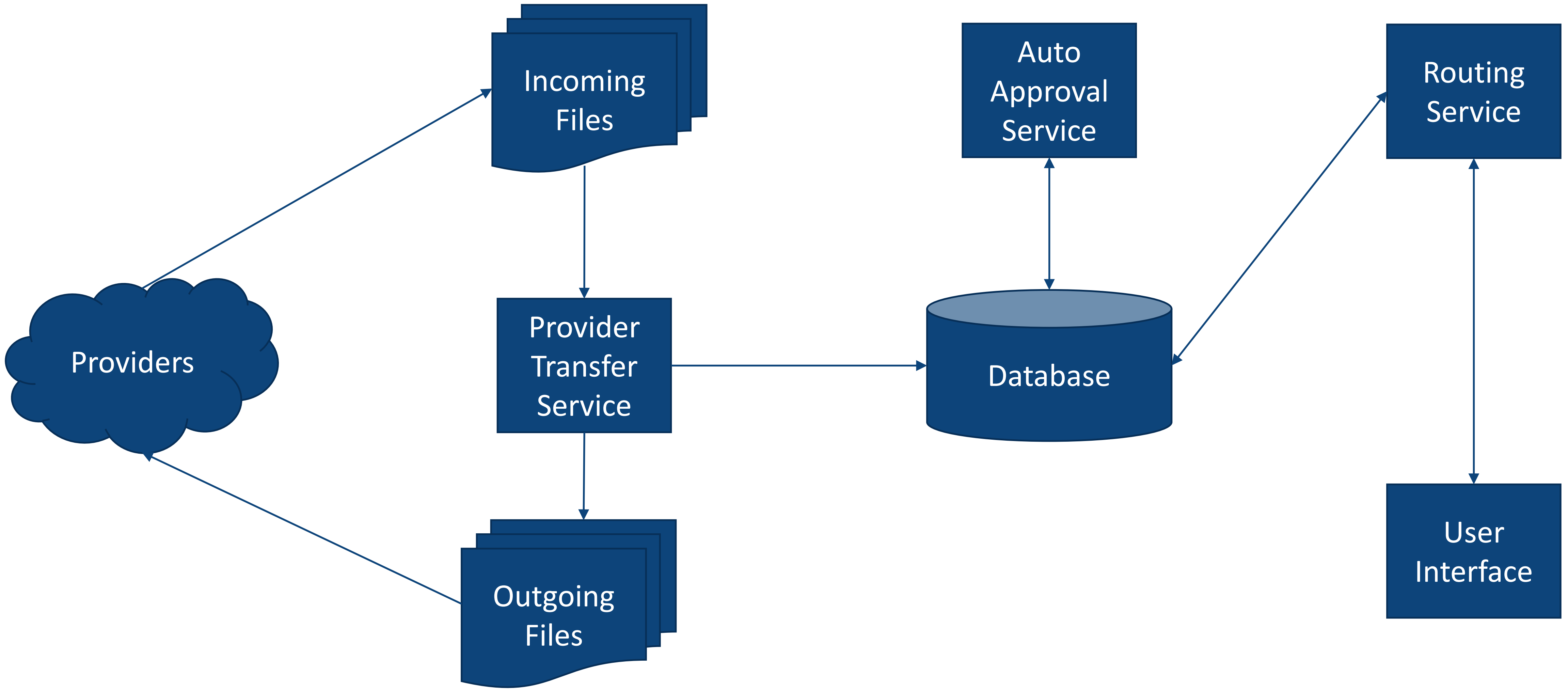
Three Amigo's Health provides utilization management reviews for health care insurers. The insurers that employ TAH require providers to get approval before administering sleep studies, fertility treatments, or long-term care to their patients. This process helps ensure that the services are appropriate for the patient at the time.

The process starts when the providers send their requests to TAH. TAH has a process to automatically approve many of the requests, but those that can't be automatically approved are routed to a TAH physician for a decision. Once a decision is made, the decision is sent back to the provider. The providers usually get a decision in 1 to 3 days.

Process Flow



Current Solution



Current Issues

THREE AMIGO'S HEALTH

- Solution not keeping pace with rapidly growing business
  - Users receive timeout errors during busy times
  - Lead time for new hardware limiting system's ability to scale
    - Management hesitant to let servers to sit idle during non-peak times
  - Team does not have skills to manage network load balancing

- Turnaround time for providers needs to be reduced from days to hours
  - Input/output from/to providers only done in nightly jobs

- New features and enhancements take too long or not possible
  - Need real-time view of active requests, but additional load likely to lead to additional performance issues

Goals of Event-Driven System

- Performance / Scalability
  - Eliminate sources of timeout issues
  - Scale hardware to meet peak demands without wasting resources during non-peak times
    - Avoid lead-time required for new hardware
  - Enable near real-time view of active messages
  - Allow processing of requests as received and returning decisions as soon as made
  - Continue to scale with business

- Must be easy to add new features
  - Decouple components and teams

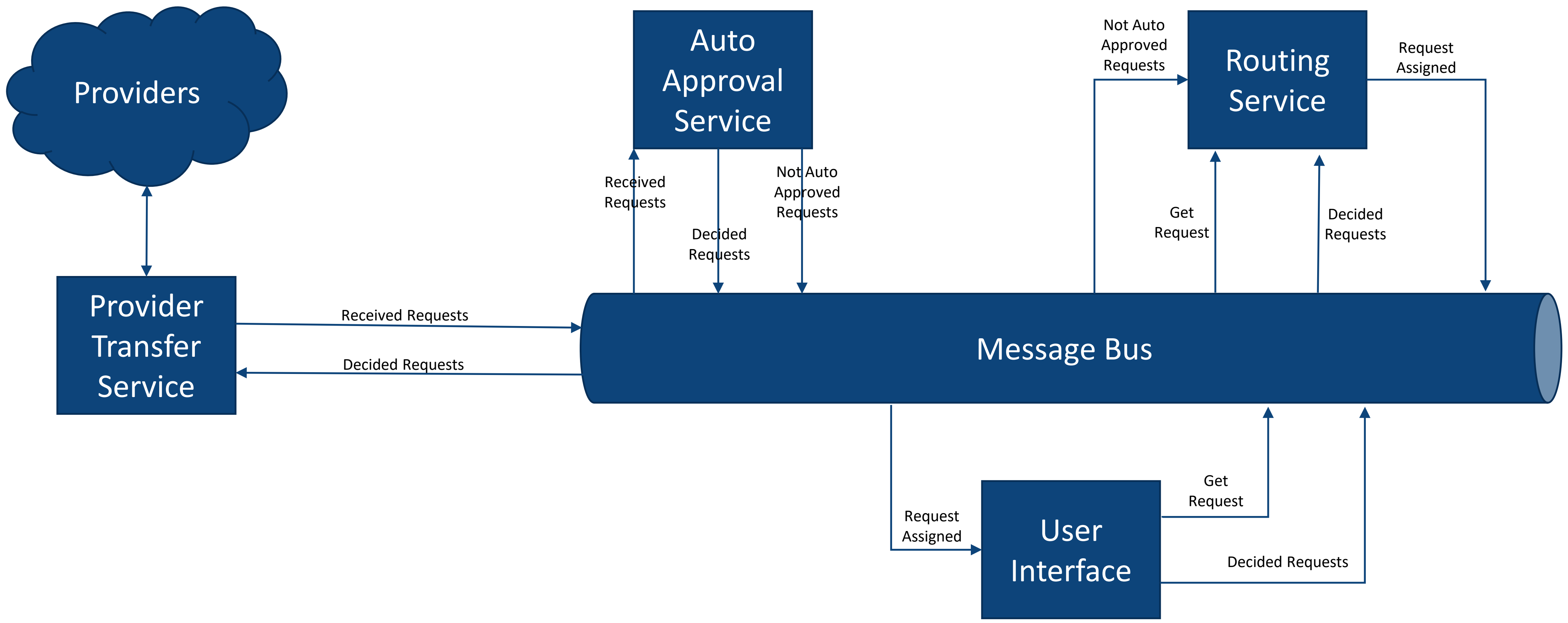
Approach for Event-Driven System

THREE AMIGO'S HEALTH

- Take advantage of cloud platform, such as Azure
  - Instances can be added/removed dynamically (in minutes) based on load
  - Workload will adjust seamlessly as components added/removed
- All system changes published as events via asynchronous message bus
  - Event: notification that something has occurred
  - Publishers send messages to a message bus
  - Consumers receive messages from the message bus
- Central database replaced with message bus and local repositories
  - Serves events, including historical events, to any component at any time

Final Solution

THREE AMIGO'S HEALTH



Providers
Provider Transfer Service
Received Requests
Decided Requests
Auto Approval Service
Received Requests
Decided Requests
Not Auto Approved Requests
Message Bus
Not Auto Approved Requests
Routing Service
Request Assigned
Get Request
Decided Requests
Request Assigned
User Interface
Get Request
Decided Requests