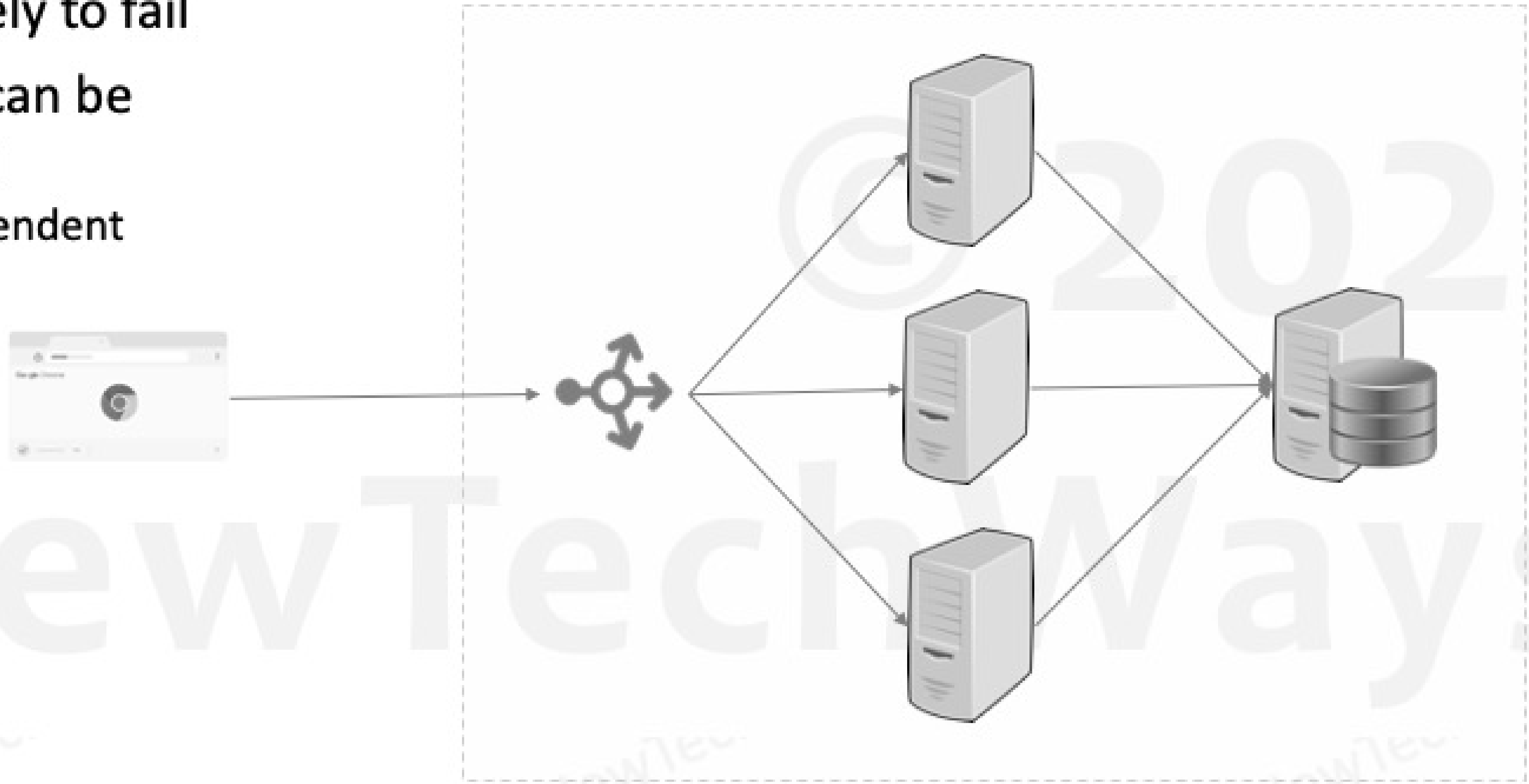
System Reliability

- Reliability
  - Availability, Reliability
  - Fault Tolerance
  - Partial Failures
- Designing Fault Tolerance
  - Redundancy
    - Hot, Warm, Cold
  - Fault Detection
    - Health Checks, Monitoring
  - Recovery
    - Stateless, Stateful
- System Stability
  - Timeouts
  - Circuit Breakers
  - Fail Fast, Shed Load

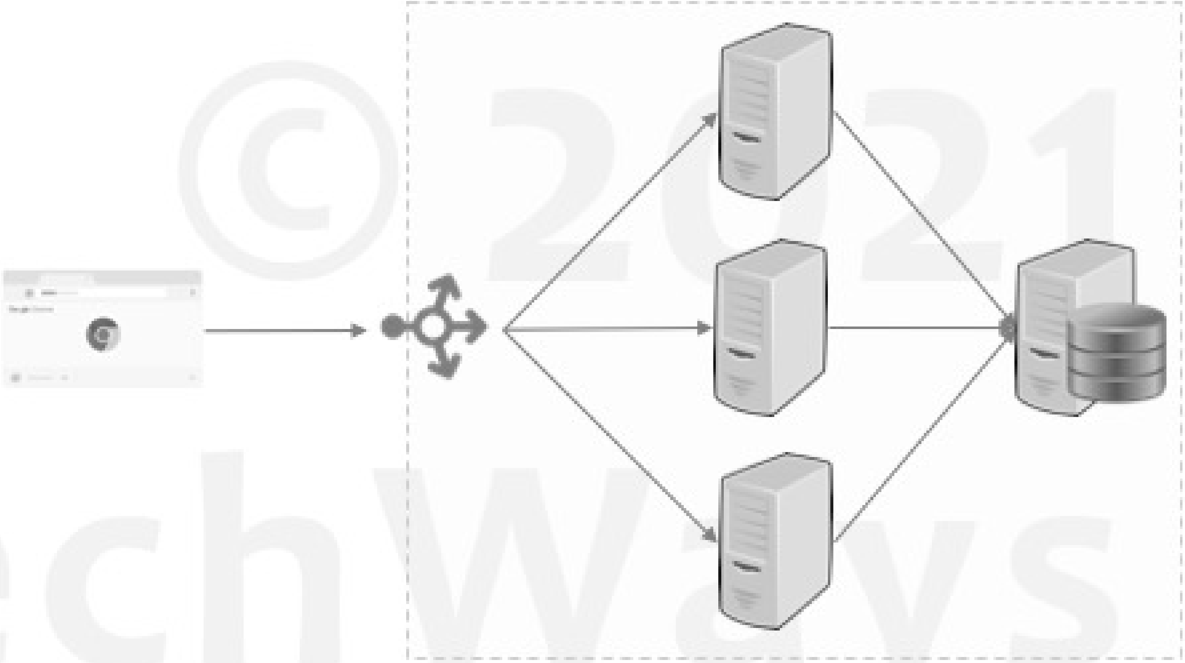
Distributed Systems

- More likely to fail
- Failures can be
  - Partial
  - Independent



Failures in Large-Scale Systems

- Large scale systems are generally distributed systems
  - Large number of components
  - Large number of component instances
- Failures can be
  - Partial
  - Independent
  - Single point of failures
- Increased chance of partial failures
- Partial failures can lead to complete system failures



Partial Failures

- Network Failure – LAN, WAN, Load Balancer
- Machine Failure – CPU, Disk, Memory
- Software Failure - Process
- Disaster – Datacenter
- Operations
  - Deployment Failure
  - Configuration Failure
  - Load Induced Failure
  - External Service Failure

After a point, its much more economical to recover from a failure instead of preventing it altogether

- No matter how hard we try
  - Hardware and Networks will fail
  - A changing Software will fail
  - Disasters will happen

Reliability Engineering

- Reliability
- Availability
- Fault Tolerance

Reliability

- A system is said to be reliable if it can continue to function correctly and remain available for operations even in the presence of partial failures.
- It is measured as the probability of a system working correctly in a given time interval

Availability

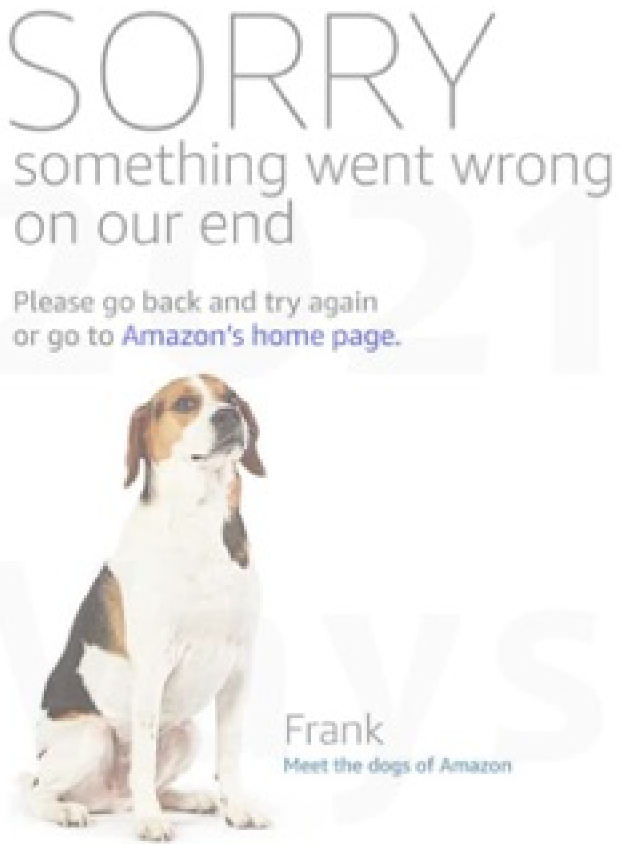
- It is the probability of a system working correctly at any given time and being available for operations
  - Time based availability

$$\text{availability} = \frac{\text{uptime}}{(\text{uptime} + \text{downtime})}$$

  - Request based availability

$$\text{availability} = \frac{\text{successful requests}}{\text{total requests}}$$

- There can be downtime but the system is expected to recover from the same in a quick time



SORRY

something went wrong on our end

Please go back and try again
or go to Amazon's home page.

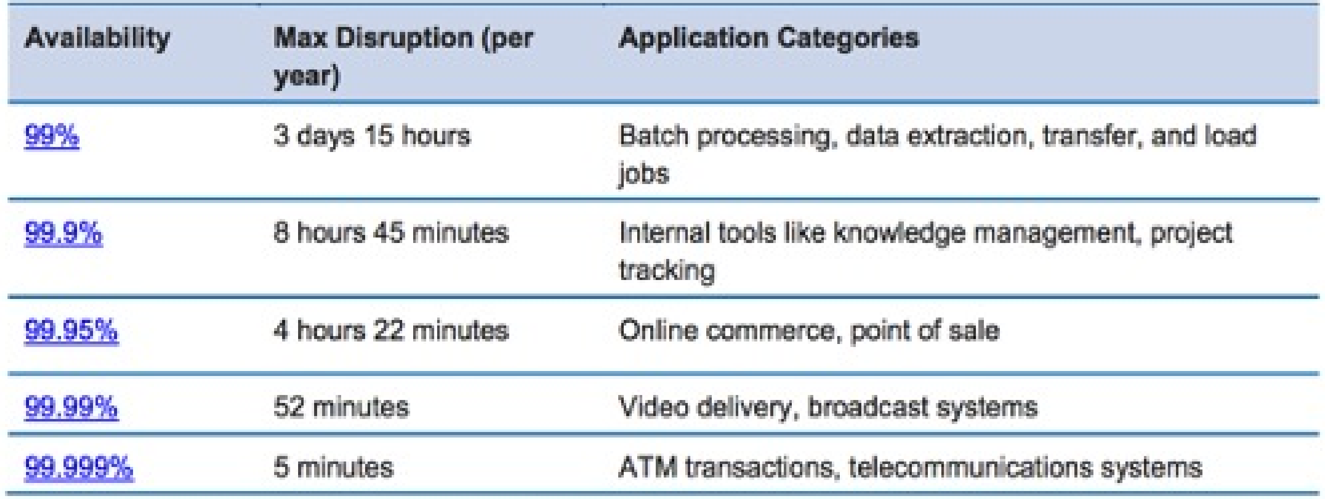
Frank

Meet the dogs of Amazon

High Availability

- Availability requirements should come from the impact of availability on a business
- Beyond business, availability is at the cost of
  - New features
  - Operational costs
- The system should use downtimes permitted by SLA/SLO for rollout of new features
  - New feature rollouts invariably cause disruptions

Availability Requirements



Availability	Max Disruption (per year)	Application Categories
99%	3 days 15 hours	Batch processing, data extraction, transfer, and load jobs
99.9%	8 hours 45 minutes	Internal tools like knowledge management, project tracking
99.95%	4 hours 22 minutes	Online commerce, point of sale
99.99%	52 minutes	Video delivery, broadcast systems
99.999%	5 minutes	ATM transactions, telecommunications systems

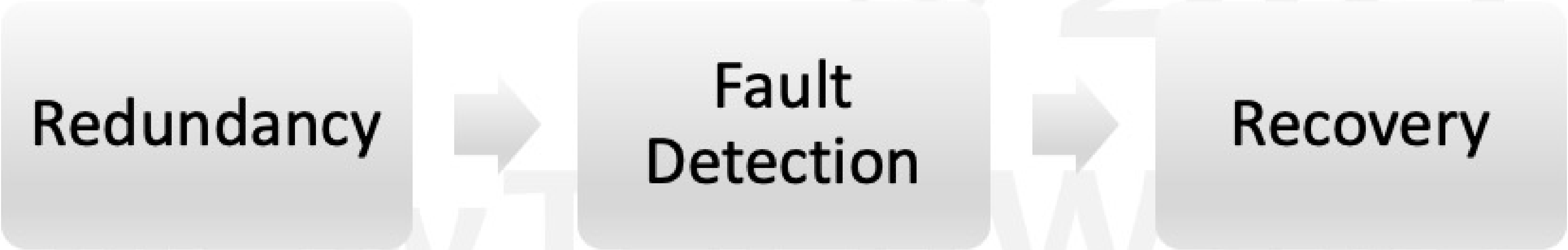
99.999% availability means almost no downtime throughout the year

Fault Tolerance

- Fault Tolerance is a technique to improve Availability and/or Reliability of a system
- It is commonly referred to as an ability of a system to automatically
  - Detect partial failures
  - Handle partial failures
  - Recover from partial failures
- Serviceability
  - The ease with which a system can be serviced in the event of a failure also determines the availability of a system

Designing Fault Tolerance

Fault Tolerant Design

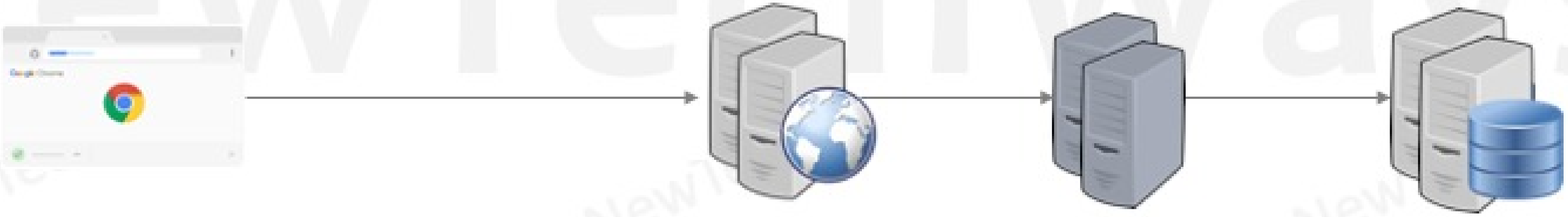

Redundancy

Fault Detection

Recovery

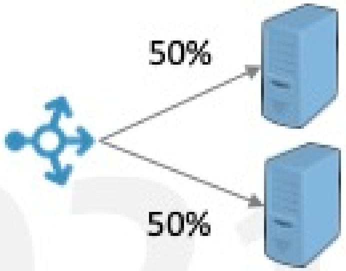
Redundancy

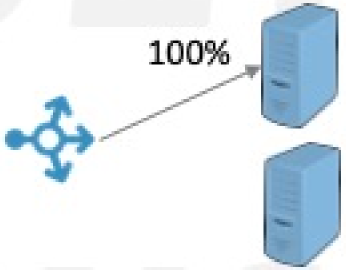
- Replication/Duplication of critical components or functions of a system in order to increase its reliability
- A secondary capacity is kept ready as a backup, over and above the primary capacity, in case the primary is not available

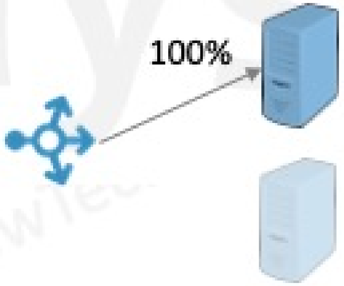


Types of Redundancy

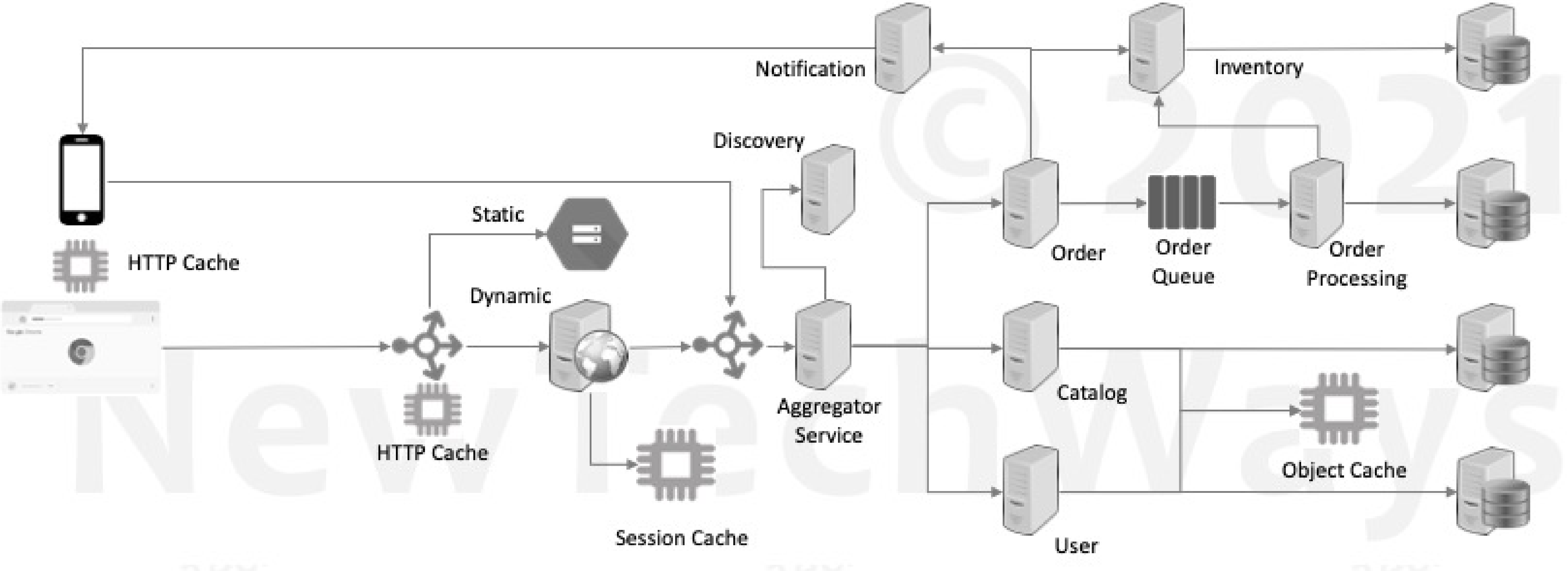
- Active Redundancy – Hot Spare
  - All nodes do the processing
  - Ideal for providing highest availability
- Passive Redundancy – Warm Spare
  - Only actives nodes do the processing
  - Ideal for quick recovery
- Cold Redundancy – Spare (Backup)
  - Spare nodes are brought up only on a failover
  - It is not a high availability option



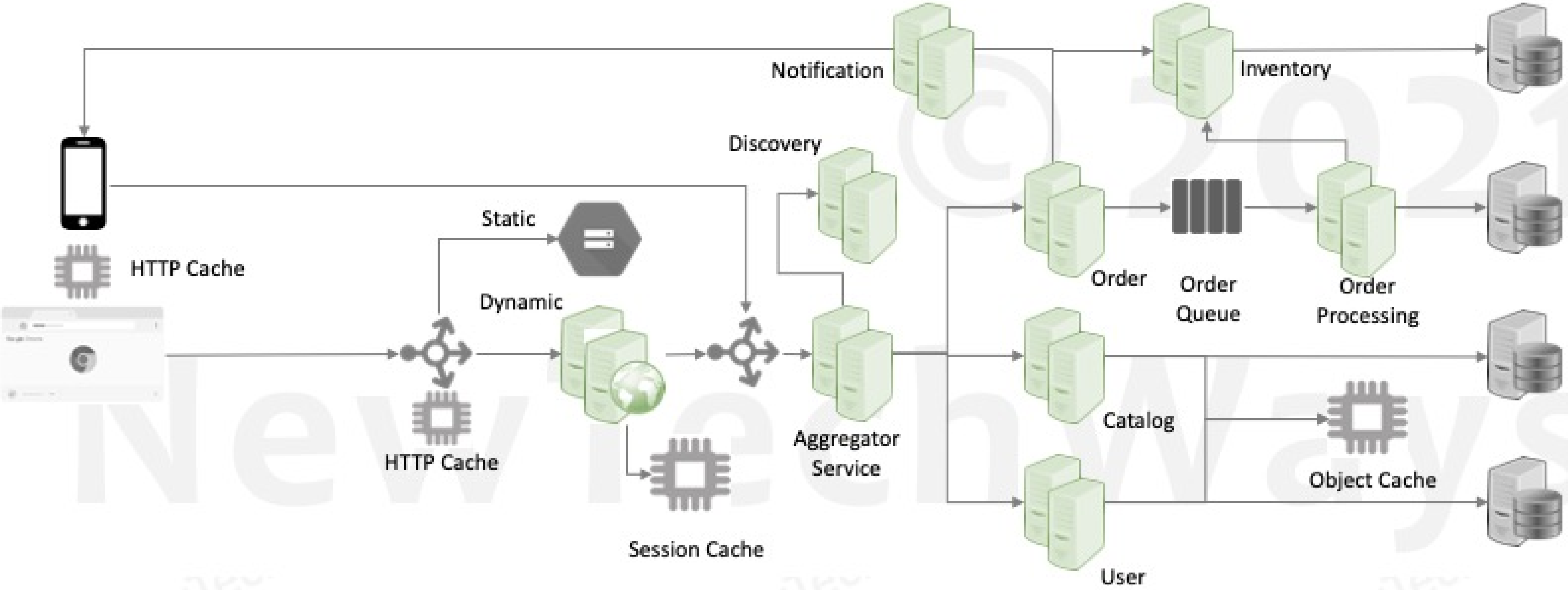




Single Points of Failure

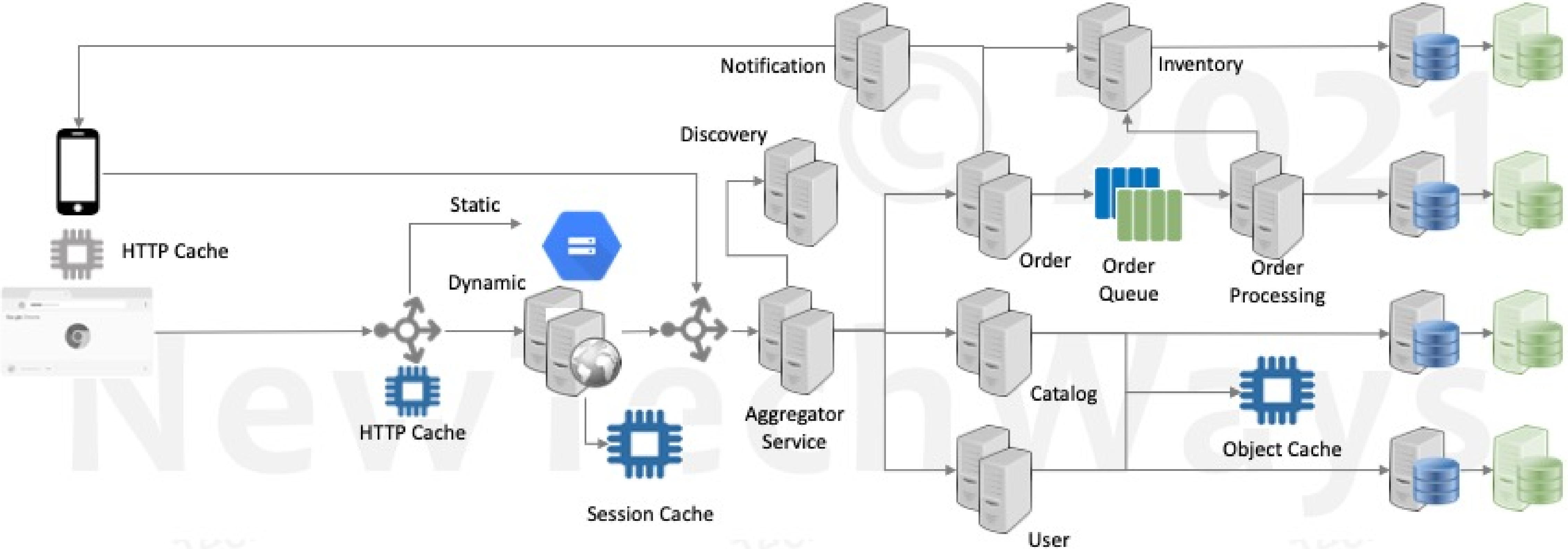


Redundancy for Stateless Components



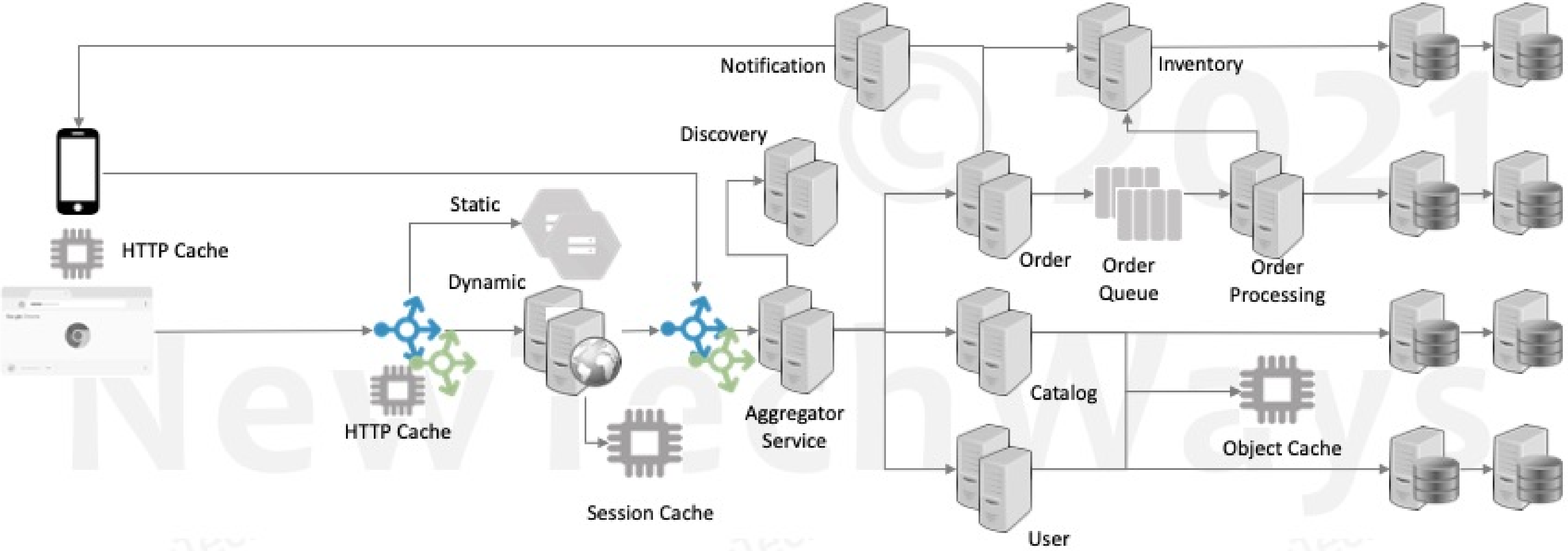
NewTechWays

Redundancy for Stateful Components

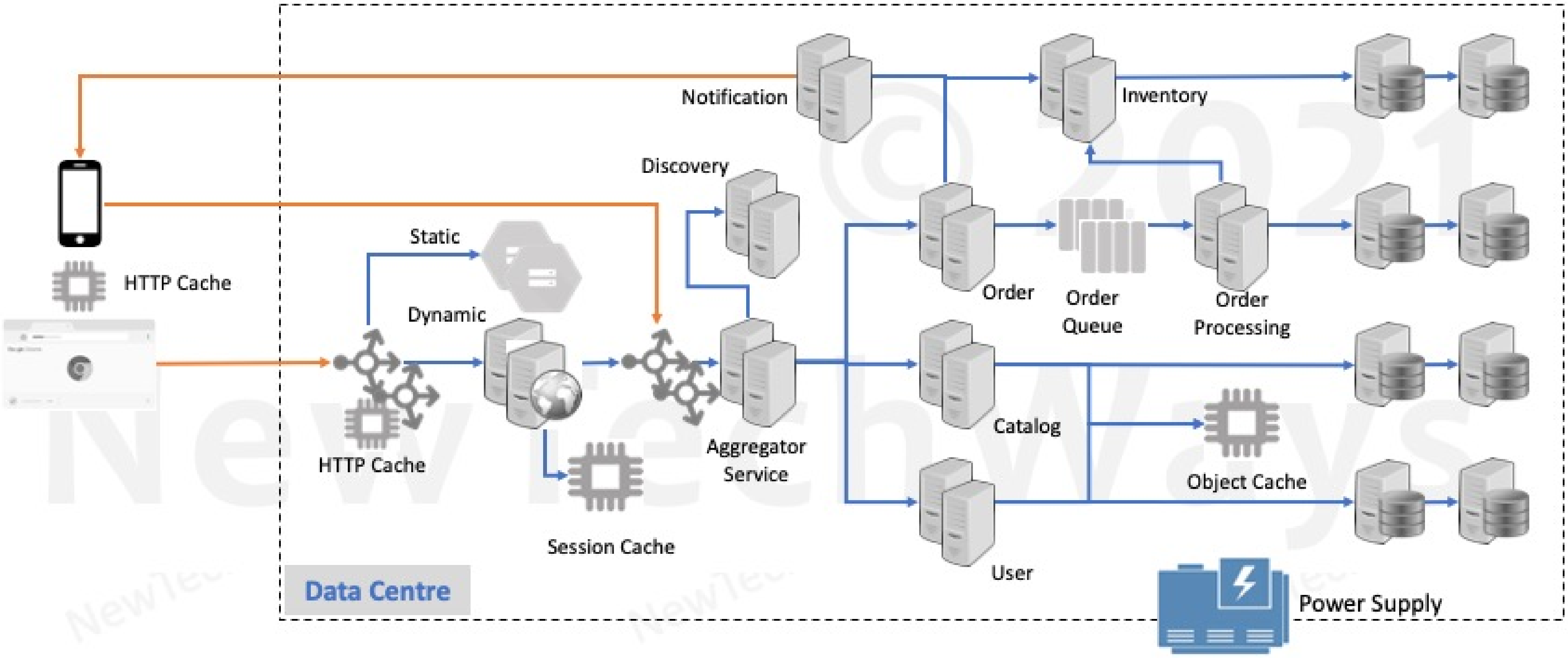


NewTechWays

Load Balancer Redundancy

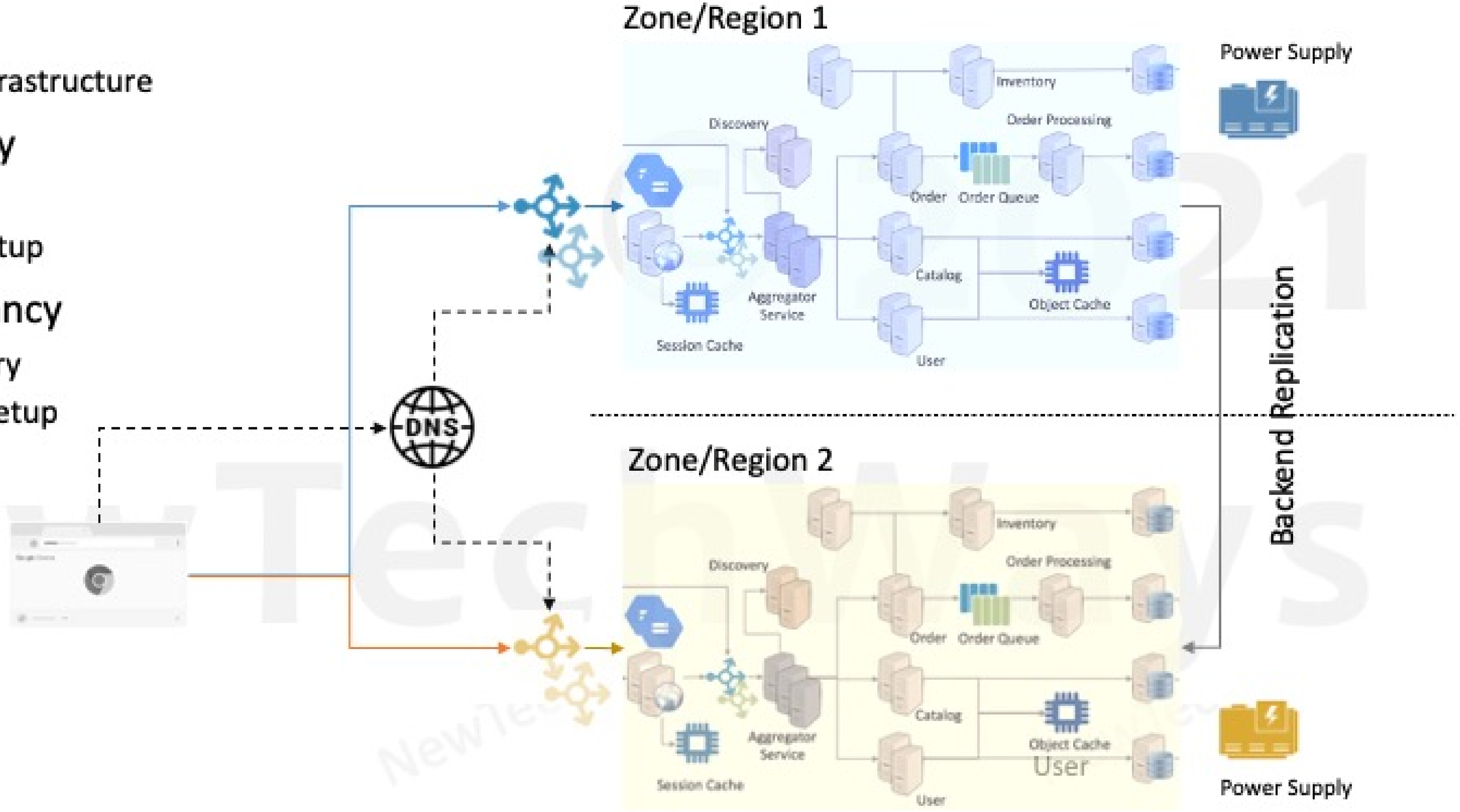


Infrastructure as SPOF

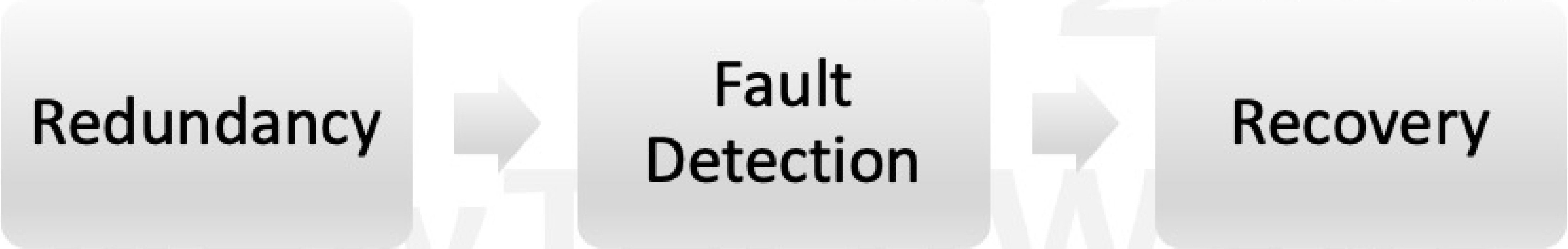


Datacenter Redundancy

- Fault Isolation
  - Independent infrastructure
- Zonal Redundancy
  - High Availability
  - Active-Active Setup
- Regional redundancy
  - Disaster Recovery
  - Active-Passive Setup



Fault Tolerant Design

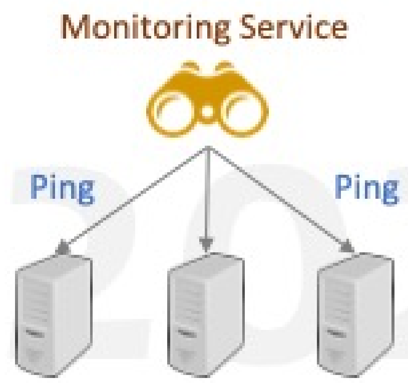


Fault Models

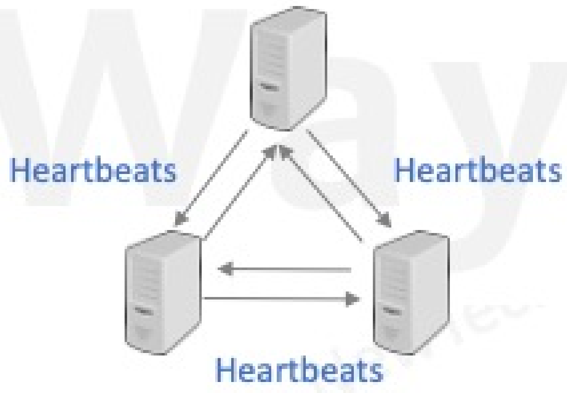
- Response Failure
  - A server fails to receive or respond to incoming messages
- Timeout Failure
  - A server response duration is longer than timeout duration
- Incorrect Response Failure
  - A server's response is incorrect
- Crash Failure
  - A server halts but is working correctly until it halts
- Arbitrary Response Failure
  - A server's response is incorrect because its security is compromised

Health Checks

- External Monitoring Service
  - Ping based

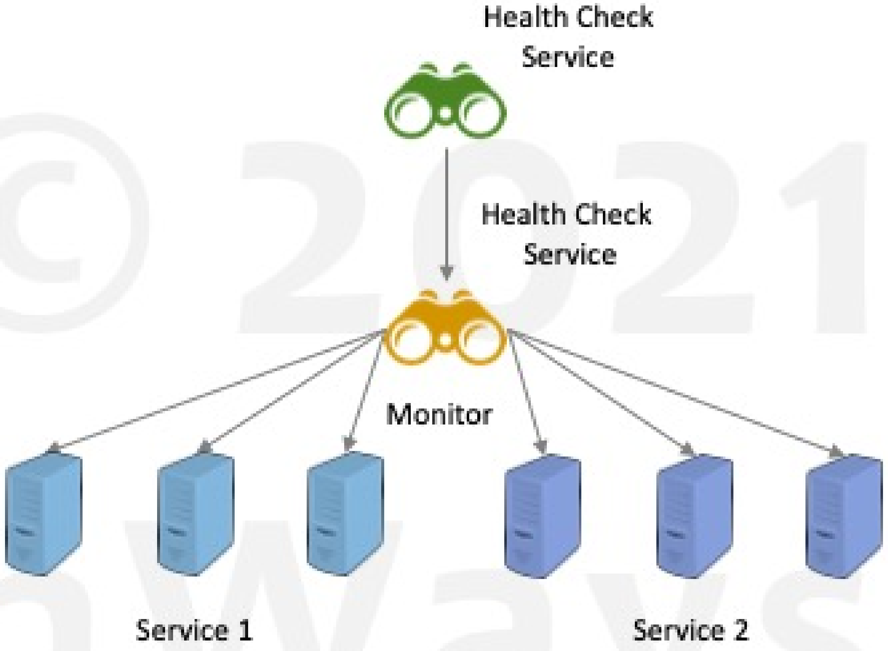


- Internal Cluster Monitoring
  - Heart-beat based



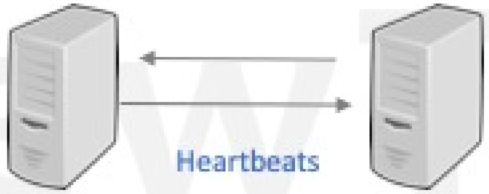
External Monitoring Service

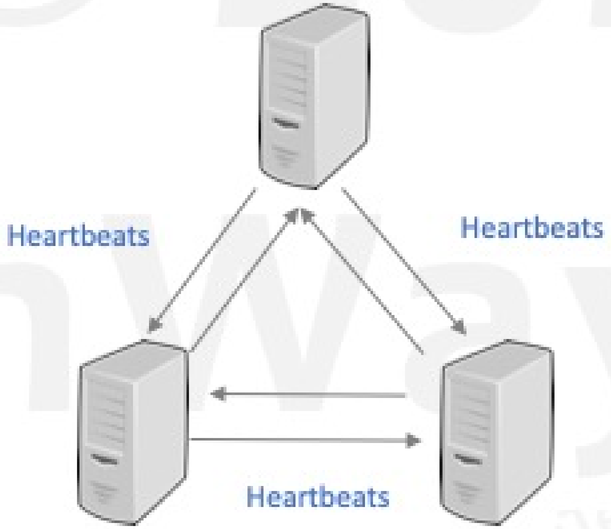
- Health check service generates
  - Alerts – for recovery
  - Events – for scaling
- Application Health Checks
  - HTTP Response
  - TCP Response
- Periodic Health Checks
  - Response Code
  - Response Time
  - Number of Retries
    - Up
    - Down



Internal Cluster Monitoring

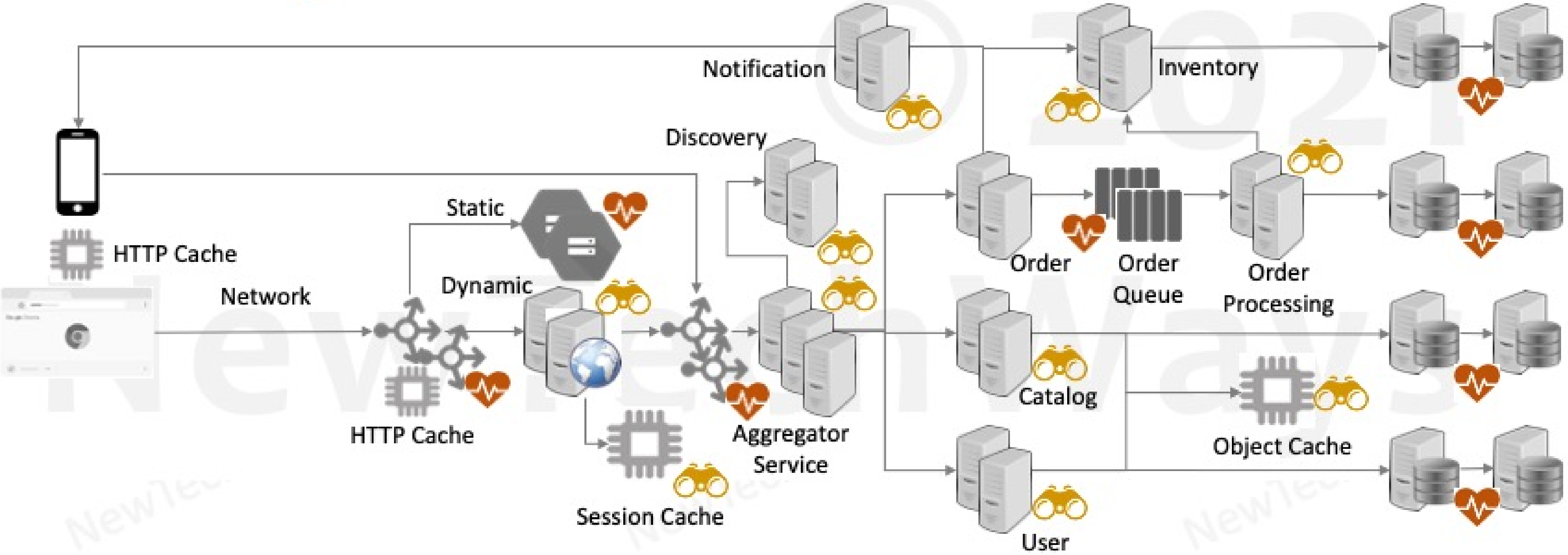
- Periodic exchange of heartbeats between redundancy cluster nodes
  - Requires protocols for communication and recovery
- Useful for stateful cluster components
  - Examples are NoSQL DB cluster and Load Balancers



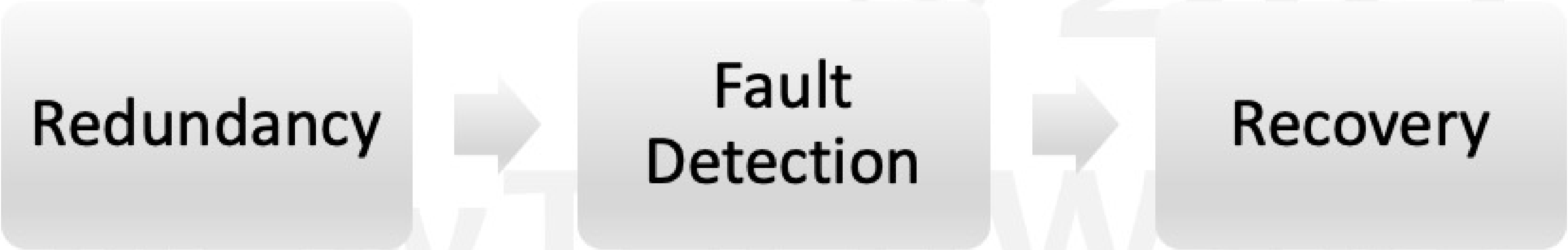


Fault Detection – Monitoring

- Health Checks
- Heart Beats

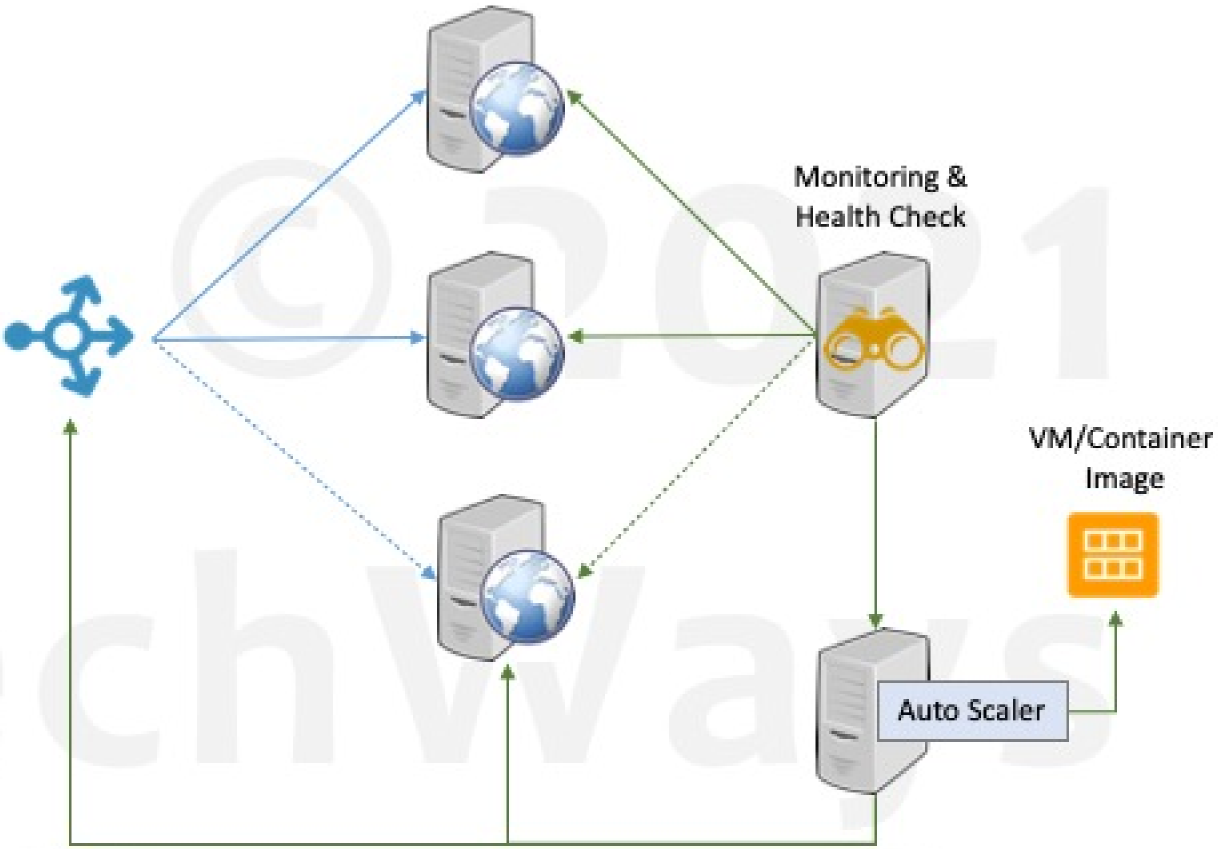


Fault Tolerant Design



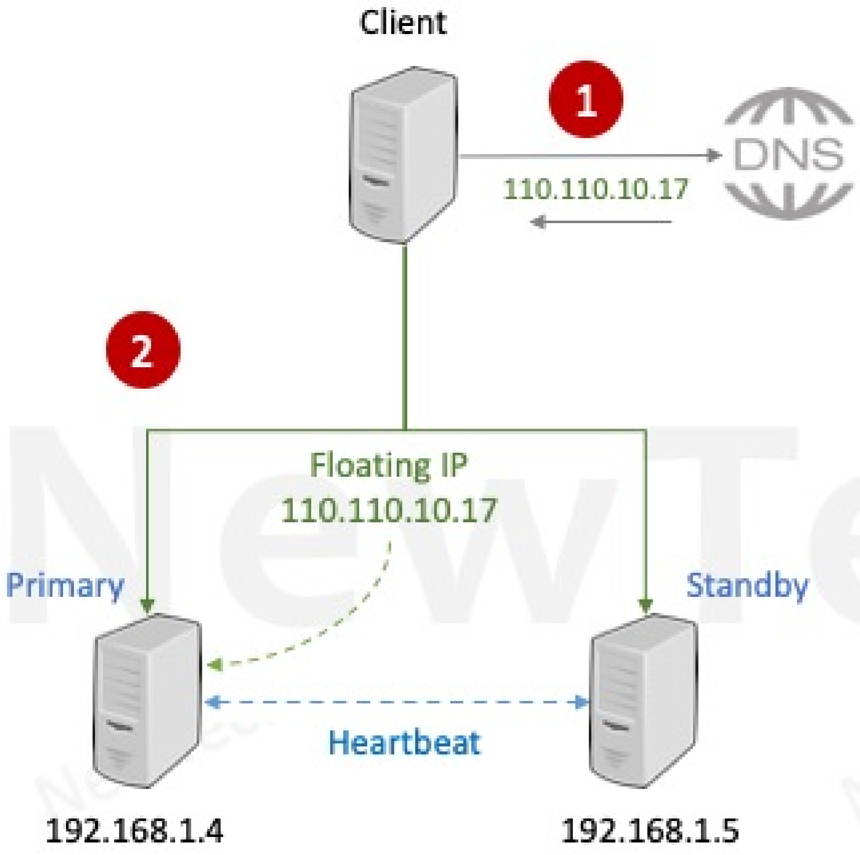
Stateless Recovery

- Can use existing scalability mechanism for recovery
- Hot standby
  - Have active redundant instances up and running
- Warm standby
  - Bring up new instances as and when needed
  - Terminate unhealthy instances if not dead already
  - Launch a new instance

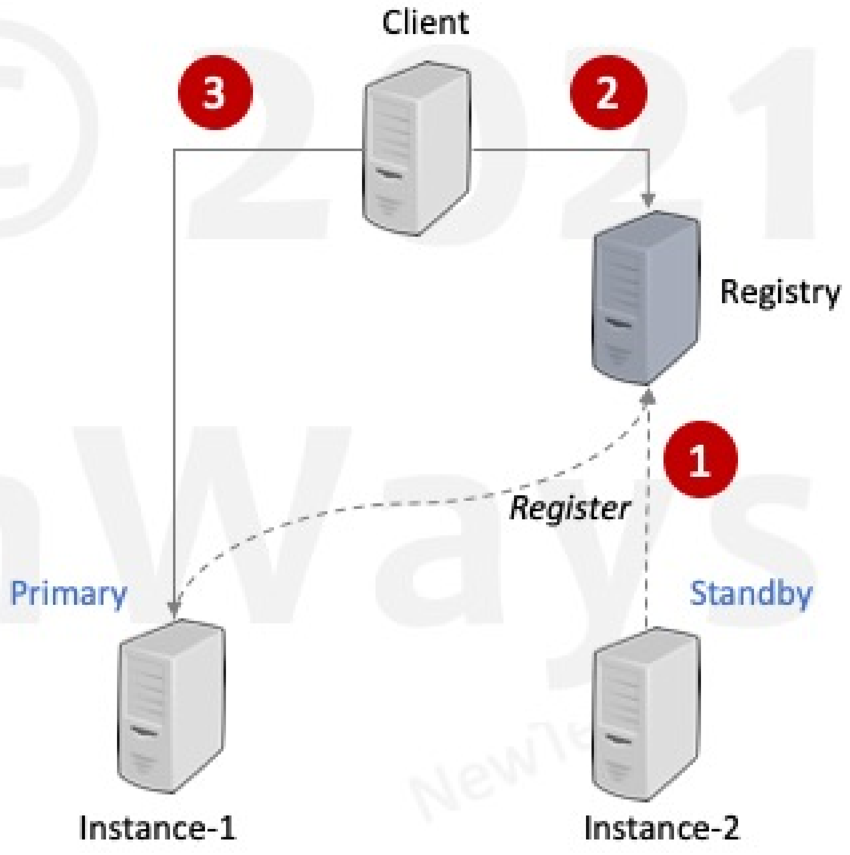


Stateful Failover

Virtual IP

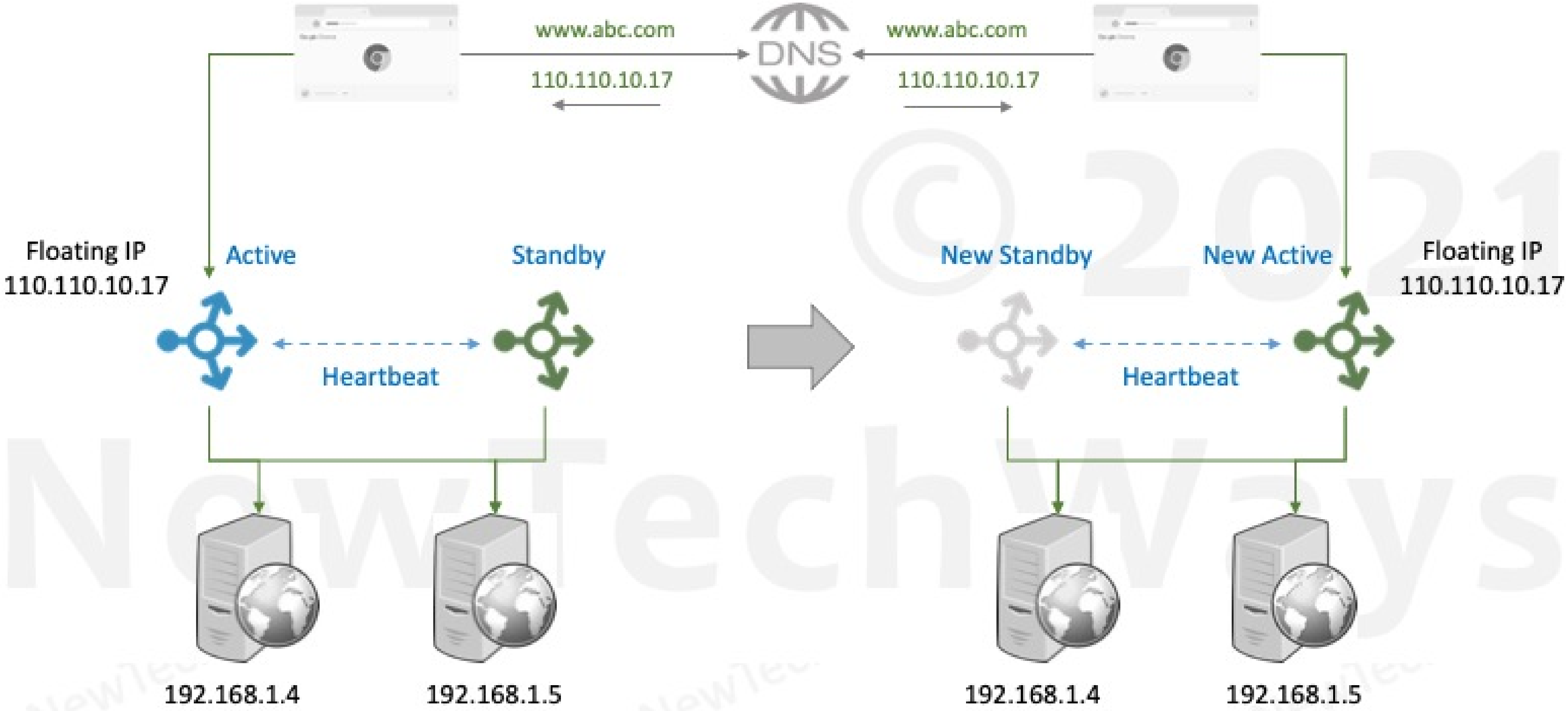


Registry/Router/DNS



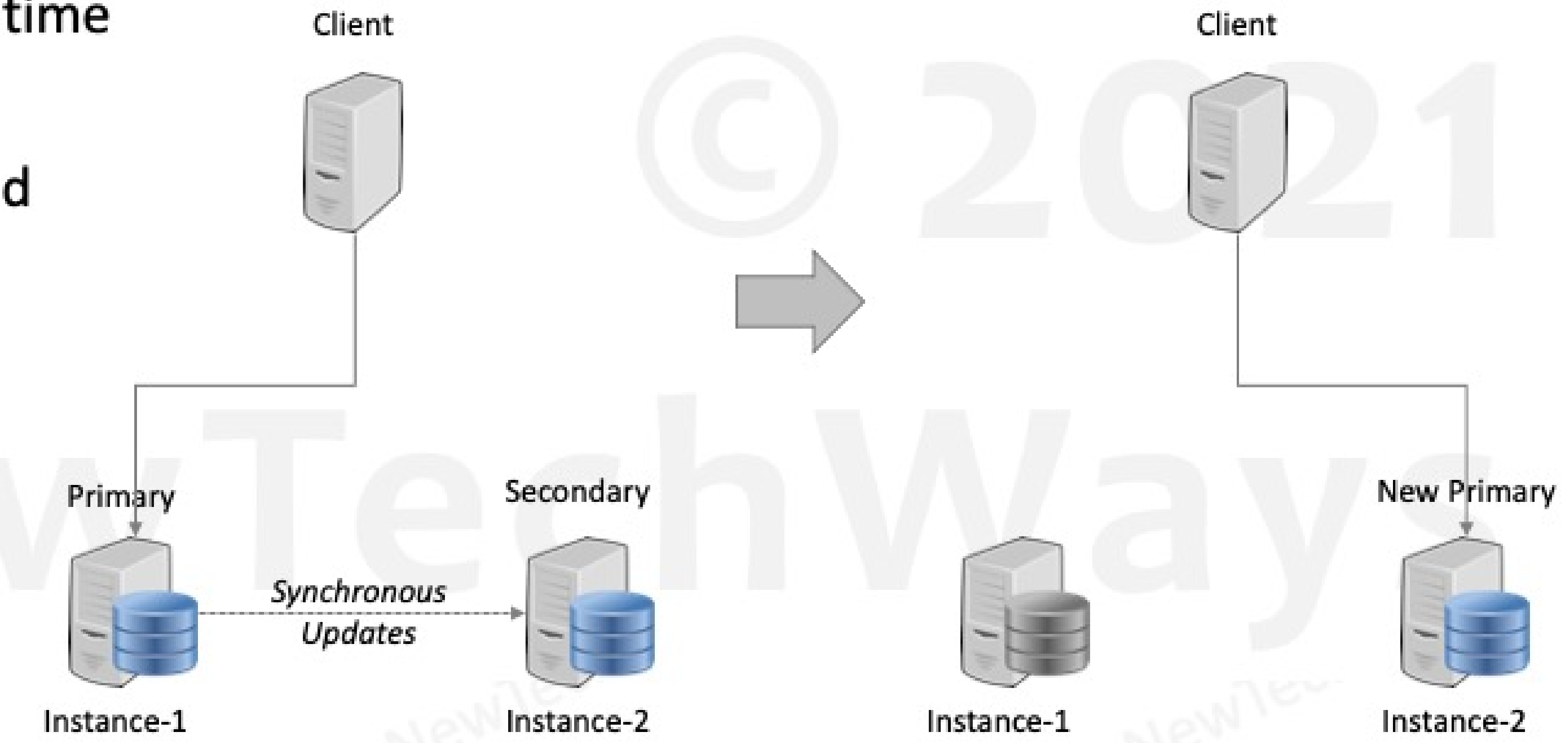
NewTechWays

Load Balancer High Availability



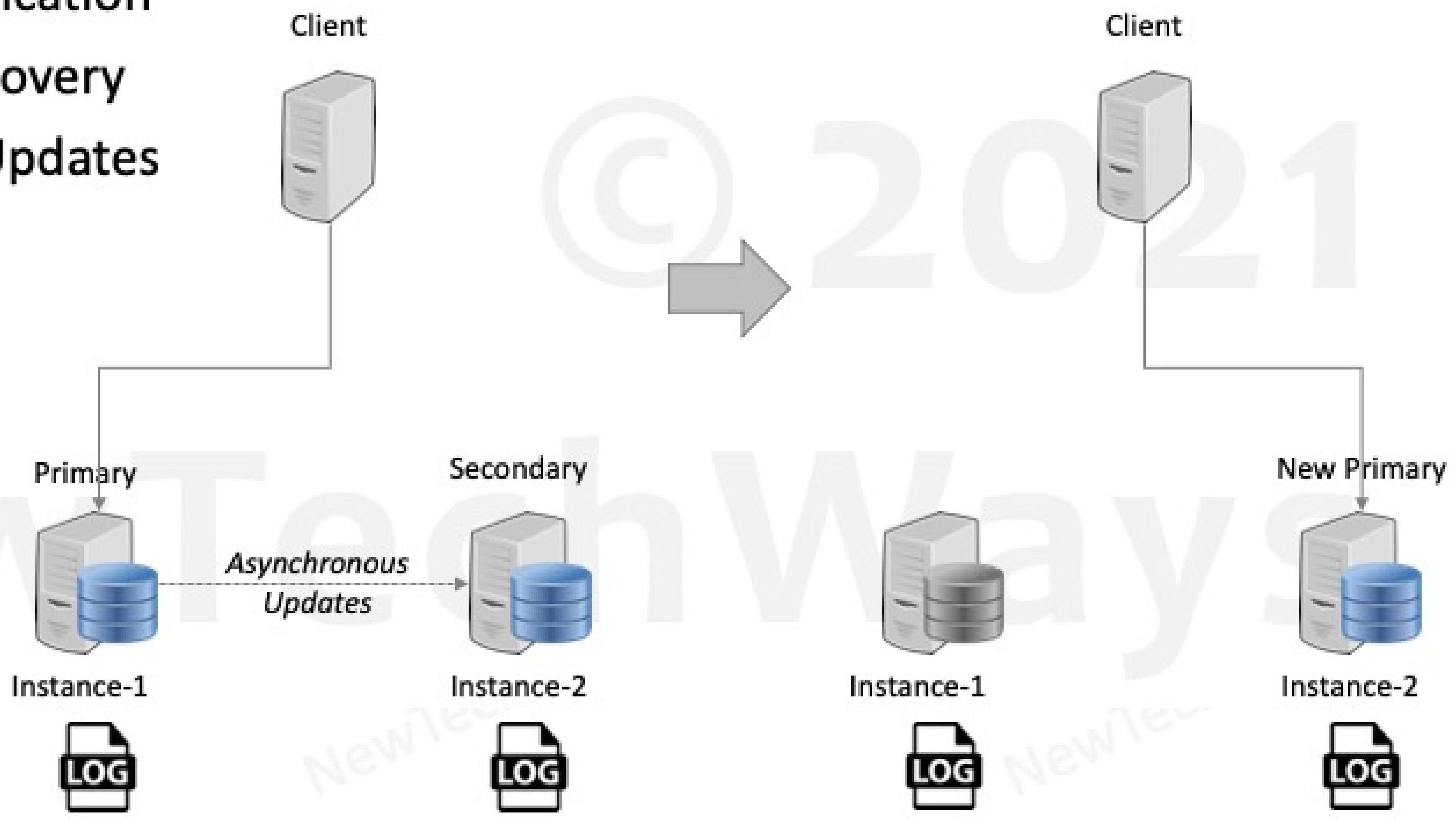
Database Recovery – Hot Standby

- Synchronous replication
- Almost no downtime
- No data loss
- Proximity needed
- Slow DB Writes



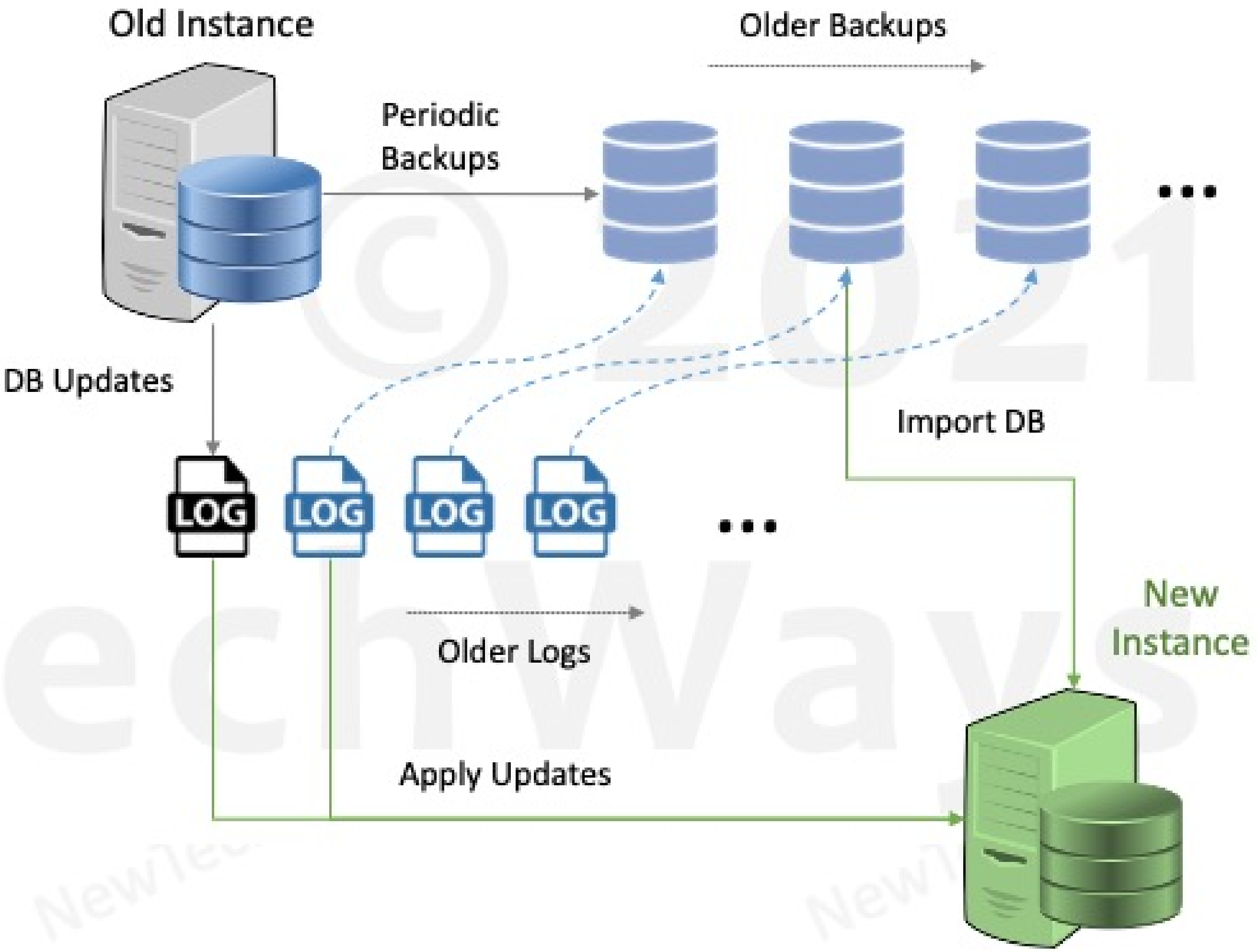
Database Recovery – Warm Standby

- Asynchronous Replication
- Catchup before recovery
- Possibility of Lost Updates
- High Performance
- Disaster Recovery



Database Recovery – Cold

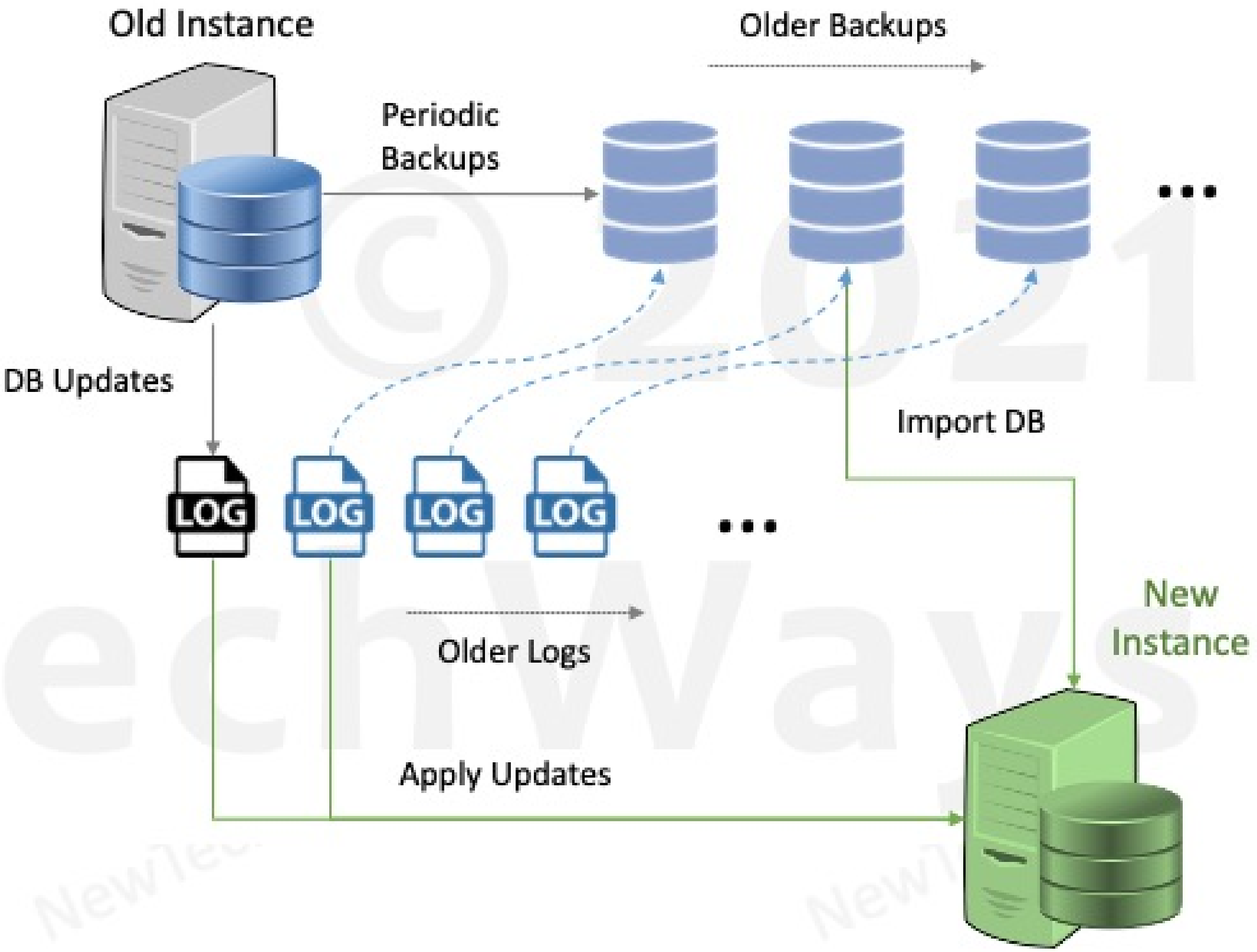
- Based on DB backups
  - Cost effective
- Significant Downtime
  - Recovery from backups
- DB Corruption
  - Replication does not help
- Process
  - Log Updates
  - Backups
    - Checkpoint
  - Import
  - Apply updates



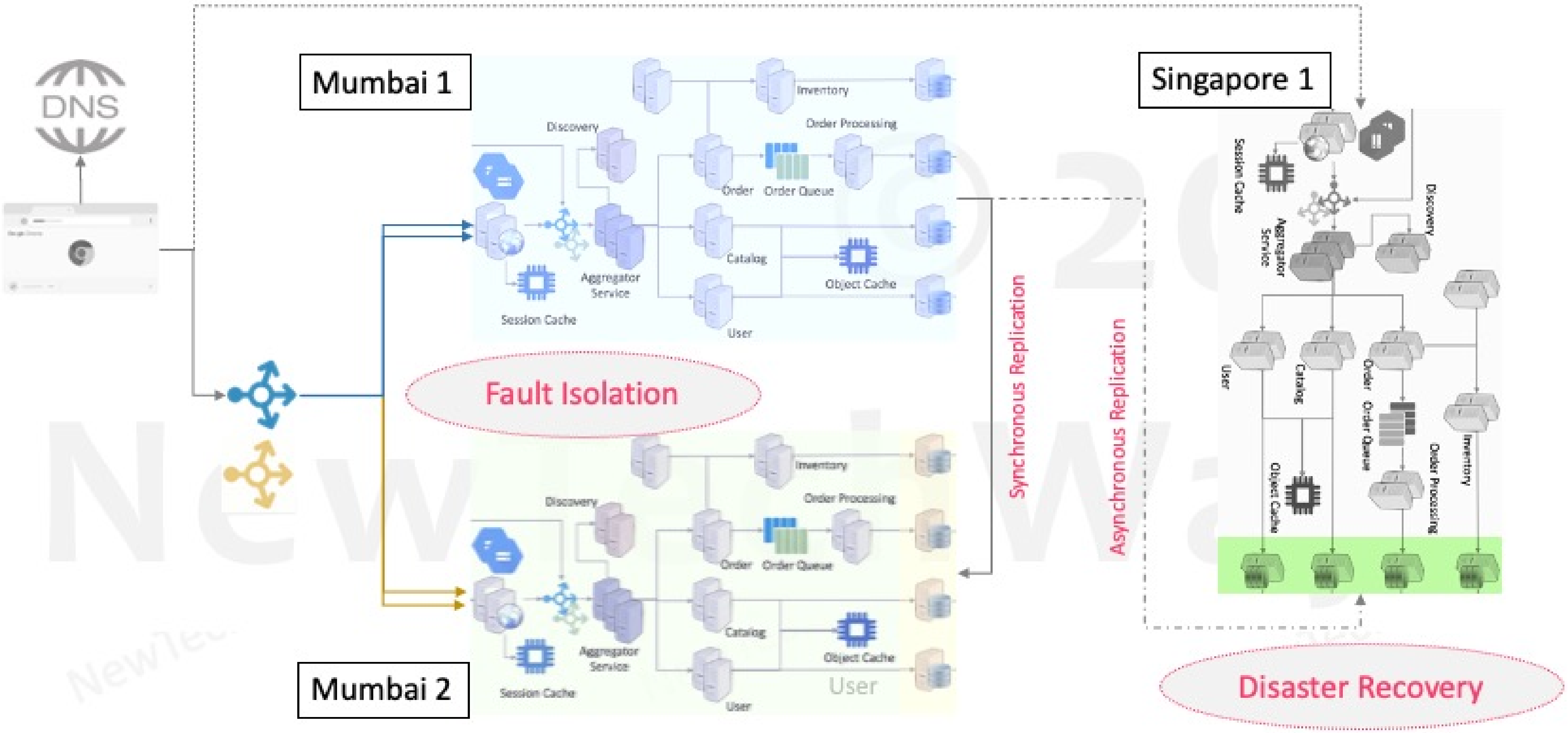
NewTechWays

Database Recovery – Cold

- Based on DB backups
  - Cost effective
- Significant Downtime
  - Recovery from backups
- DB Corruption
  - Replication does not help
- Process
  - Log Updates
  - Backups
    - Checkpoint
  - Import
  - Apply updates



High Availability in Large-Scale Systems



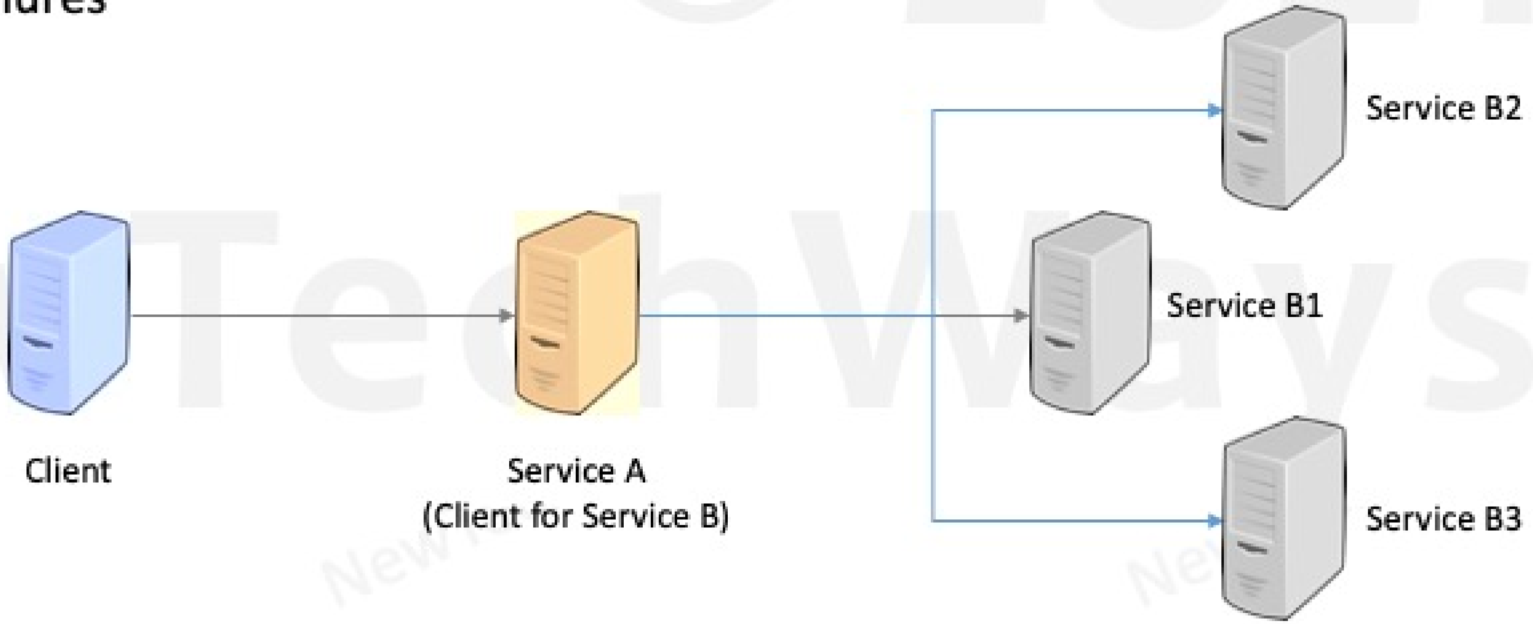
Failover Best Practices

- Failover Automation
- Regular Failover Testing in Production

System Stability

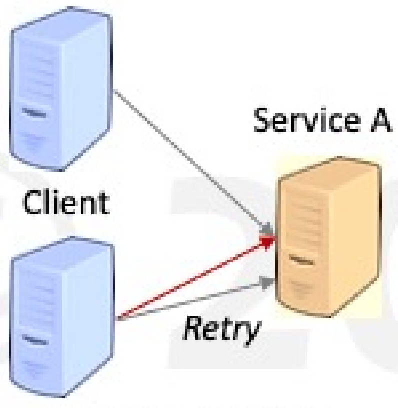
Timeouts

- Client Components
  - User interface
  - Service clients
- Timeouts prevents call to integration points from becoming blocked threads
  - Averts cascading failures

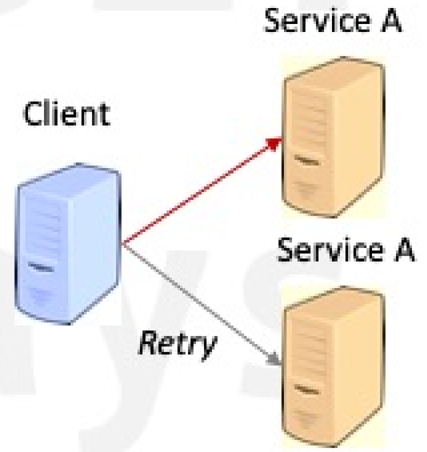


Retries

- Client Components
- For transient failures
  - Not for permanent failures
- For system errors
  - Not for application errors
- Retries with exponential back-off
- Return HTTP 503
  - Clients can decide if and when to callback again
- Use Idempotent Tokens
  - For unacknowledged failed requests
  - At least once guarantee instead of exactly once



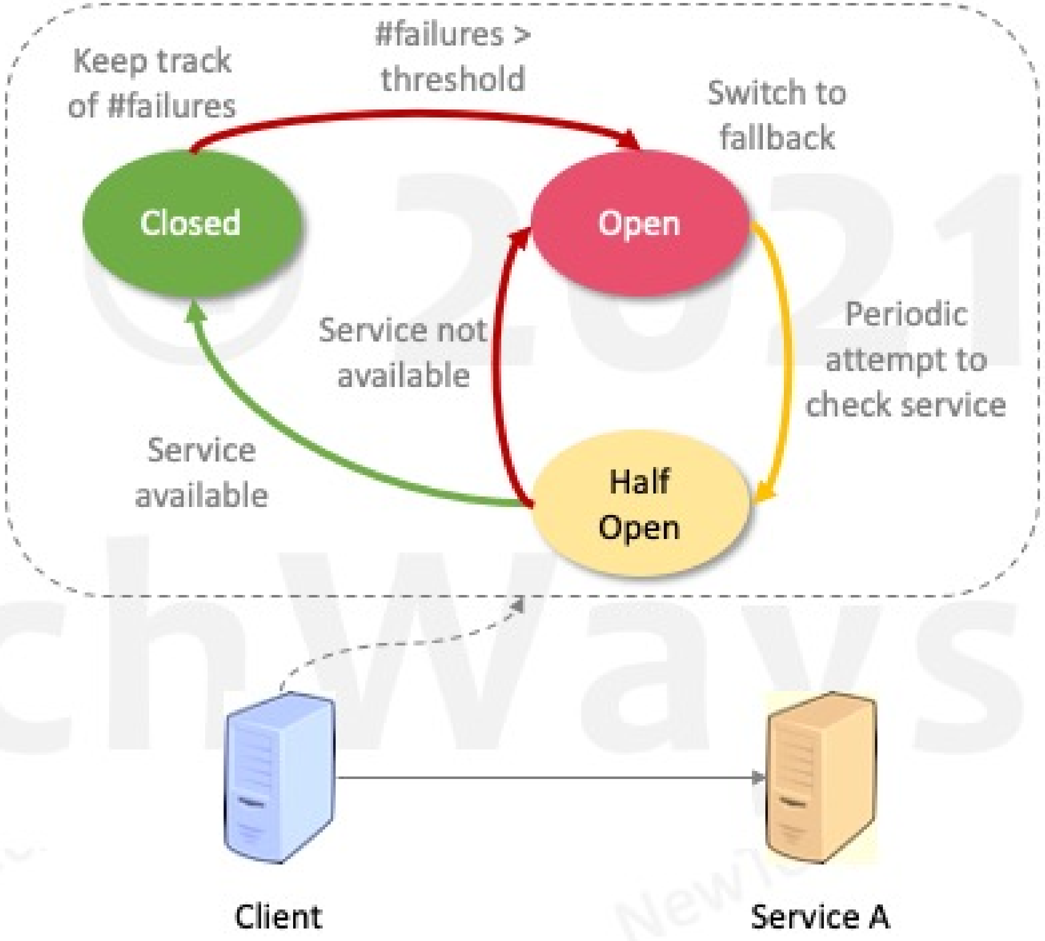
Transient Failure
Race Condition



System Error
Instance Unavailability

Circuit Breaker

- Client Components
- Deliberate service degradation when a system is under stress and a problem is detected
- Process
  - Keep track of success and failures
  - In the event of too many failures, fallback to
    - Default values
    - Cached values
    - Error messages
  - Resume when stress dissipates



Fail Fast & Shed Load

- Server Components
- Fail Fast
  - Triggered due component’s inability to process any request
    - Validation error
    - Missing Parameters/Env Vars
    - Service Timeouts (When Circuit Breaker is open)
  - Return error as soon as a component discovers it
- Shed Load
  - Failing fast due to external load on a system as a result of which excess requests cannot be processed
    - Concurrency Limits – Threads, Connections, Request Count
    - SLAs – If SLAs are not met, block/reject incoming requests
- Back Pressure
  - Shedding load for slowing down clients within a system boundary

Client
Service

NewTechWays

Summary

- Highly Available & Highly Reliable systems are Fault-Tolerant by design
- Fault tolerance is achieved by
  - Provisioning redundancy for every SPOF
    - Hot/Active, Warm/Passive, Cold/Backups
    - Stateless redundancy & Stateful redundancy
  - Building automated mechanism to detect faults
  - Building automated failover mechanism to recover from faults
    - Failover of stateless components
    - Failover of stateful components
- Stability patterns
  - Clients – Timeouts, Retries, Circuit Breaker
  - Server – Fail Fast, Shed Load, Back-pressure

Thanks!

NewTechWays

https://www.newtechways.com