MIP

A SOCIAL SOLUTION IN A DIGITAL DIMENSION

PUG CHALLENGE

"Road to High Availability begins with the Database"

Pieter J. Meyer

Introduction

- Been with MIP Holdings for over 25 years
- Head of DevOps, responsible for the oversight of Software Development and Technical Operational Standards across MIP's teams.
- Worked with Progress and their 'continually-evolving' suite of products for 30+ years.
- Knowledge and areas of expertise extends to most of the Progress OpenEdge products
- Very capable Progress OpenEdge – ABL Developer, Database Administrator and all-round solutions architect and technical consultant, … and then some more.

Pieter J. Meyer

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Agenda

PUG CHALLENGE

- Overview
- Introducing MIP
- High Availability
  - Disaster Recovery
  - Load balancing
  - Backups
  - After-Image
  - Replication
  - Replication-Sets
  - Alternate Database Connections
  - Considerations

Overview

The ultimate goal is to have your system, application and data, available 24/7, even in the event of a disaster.

At the core of any system is the data, and to achieve High Availability the road starts with the database.

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Overview

PUG CHALLENGE

High Availability is not something that is available out-of-the box with Progress OpenEdge, as both your application and database(s) needs to cater for this.

OE Replication is the first step in achieving zero-data loss, and I'll go through the different options available and methods of implementation, to suite the desired solution.

But connecting and access to the **Correct**, **Active** and **Updatable** database comes with its own challenges.

We will share our trials and tribulations of what works and what does not.
All whist keeping database security, features and connectivity in mind…

Overview

In this session I will through some of the Progress OpenEdge tools we used and tried in our quest to achieve this goal.

- OE Replication / Sets
- Alternate database connections
- etc.

All whist keeping database security, features and connectivity in mind, i.e

- CDC
- TDE
- Database Hardening
- Database Auditing
- SQL
- PAS
- etc.

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Agenda

PUG CHALLENGE

- Overview
- Introducing MIP
- High Availability
  - Disaster Recovery
  - Load balancing
  - Backups
  - After-Image
  - Replication
  - Replication-Sets
  - Alternate Database Connections
  - Considerations

Introducing MIP

PROUDLY

South African

Progress® Partner

Approx. 600 STRONG

500 Developers

PPPP

K9 Anti-Poaching
33% Female / TEARS

22+ MILLION

Beneficiaries

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Introducing MIP

PUG CHALLENGE

HEALTHCARE, RISK & FUNDING

INDIVIDUAL LIFE & RISK

GROUP LIFE & RISK

FUNDS, BENEFITS & ANNUITIES

LENDING

TECH-OPS

RESEARCH & INNOVATION

CONTINUOUS IMPROVEMENT

BIG DATA

INTERNATIONAL & OUTSOURCE DEVELOPMENT

Various Divisions across Industries

Introducing MIP

1000+ databases across our +/-70 clients

Introducing MIP

12 COUNTRIES WITHIN AFRICA

- Botswana
- Namibia
- Zambia
- Malawi
- Ghana
- Kenya
- South Africa
- Eswatini
- Lesotho
- Uganda
- Nigeria
- DRC

5 COUNTRIES OFFSHORE

- United Kingdom
- Hong Kong
- Germany
- Australia
- Canada

Across the World

Introducing MIP

- Available Skills
  - Legacy OpenEdge CHUI and GUI
  - Modern Web Systems
  - Maintenance & Support
  - Enhancements & Modernization
  - New System Development
- Outsourcing to customers in…
  - United Kingdom
  - Hong Kong
  - Germany
  - Australia
  - Canada
  - South Africa

INTERNATIONAL & OUTSOURCE DEVELOPMENT

Across In-House & Outsource

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

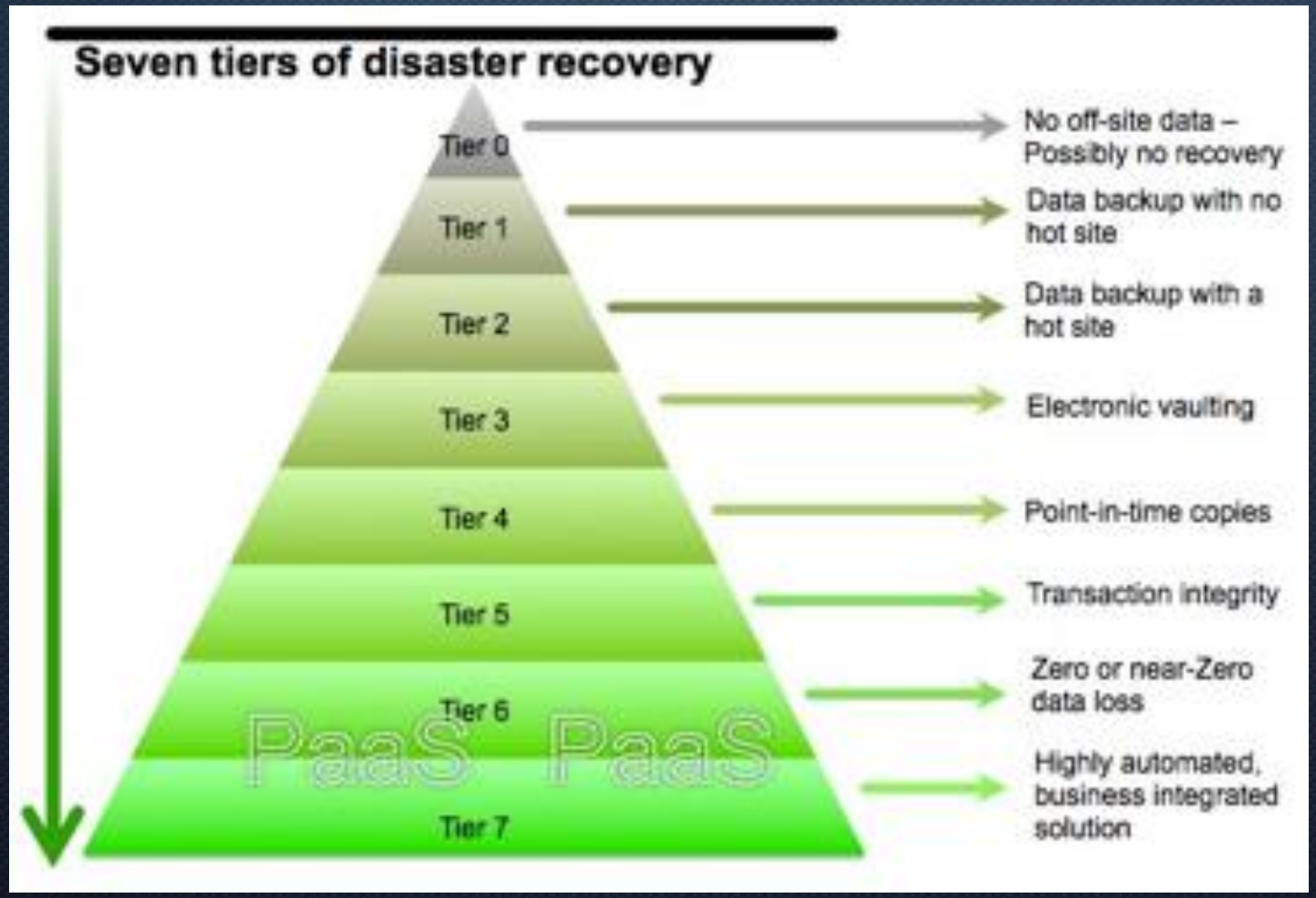
Agenda

PUG CHALLENGE

- Overview
- Introducing MIP
- High Availability
  - Disaster Recovery
  - Load balancing
  - Backups
  - After-Image
  - Replication
  - Replication-Sets
  - Alternate Database Connections
  - Considerations

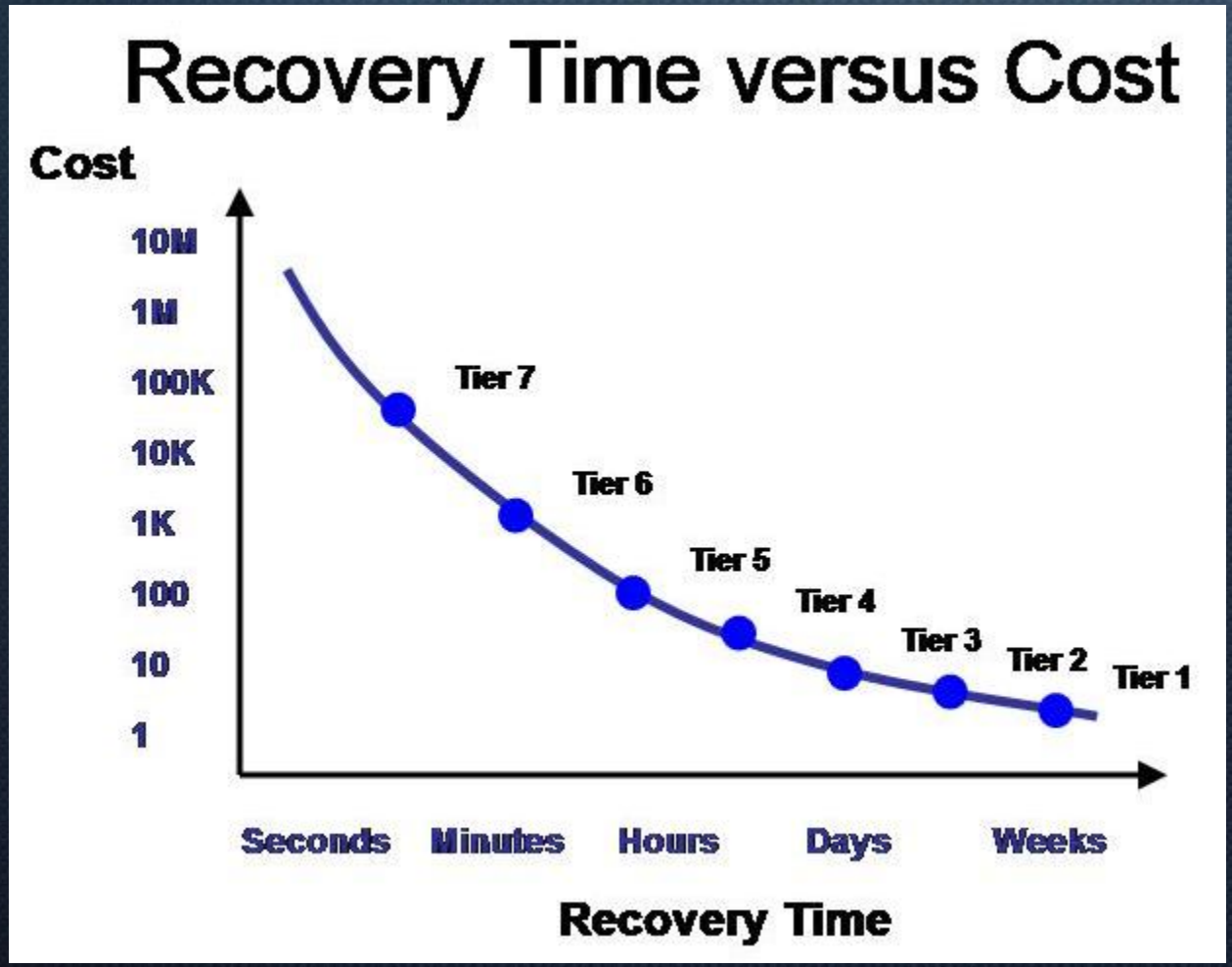
Disaster Recovery

Tiers



Disaster Recovery

Considerations

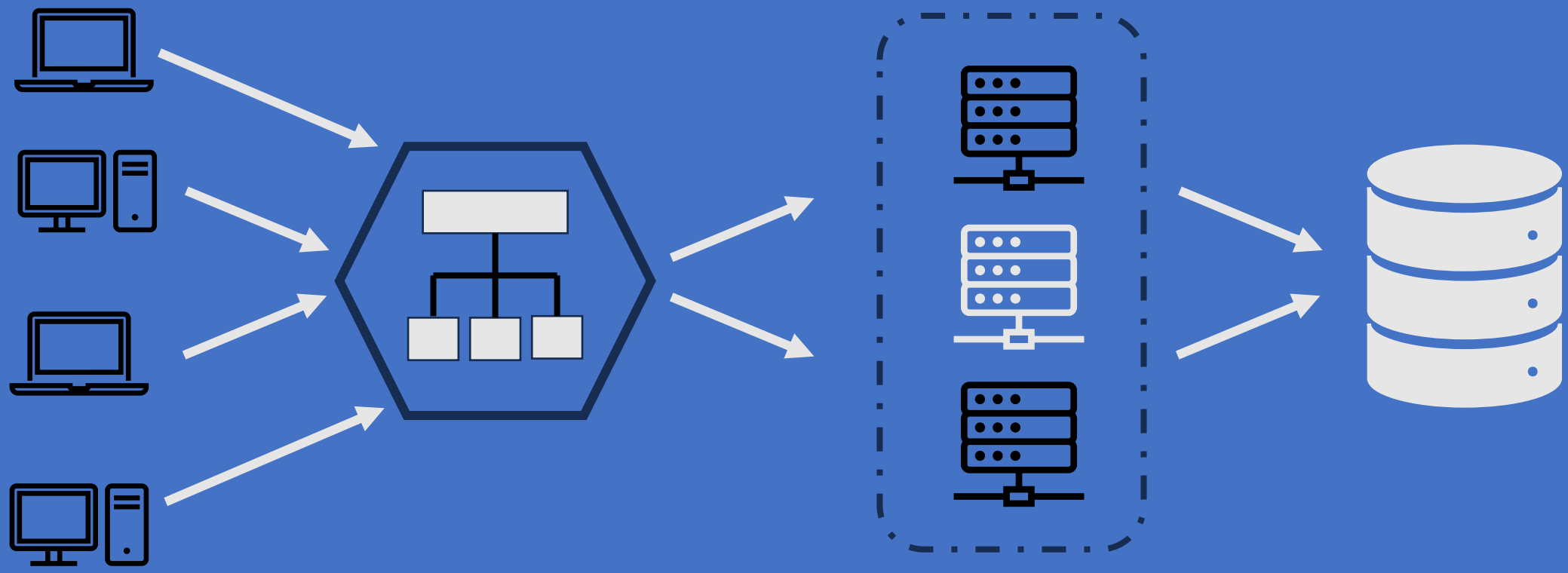


MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Load Balancing

PUG CHALLENGE

Clients Load Balancer Servers/Services Databases



Load balancing is a topic of its own

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Agenda

PUG CHALLENGE

- Overview
- Introducing MIP
- High Availability
  - Disaster Recovery
  - Load balancing
  - Backups
  - After-Image
  - Replication
  - Replication-Sets
  - Alternate Database Connections
  - Considerations

Backups & After-Image

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Replication

PUG CHALLENGE

Using OpenEdge Replication, you can replicate a local OpenEdge database, known as the SOURCE database, to up to two remote OpenEdge databases, known as TARGET databases,
that are running on one or more servers and/or data centers.

This duplication allows you to keep OpenEdge databases identical while also providing a hot standby in case a database fails.

If a database does fail, a replica becomes active, ensuring that mission-critical data is available 24 hours a day, seven days a week to your users.

Replication

Database features will work across replication

Must just be enabled for both the source and target databases

- Extents added
- Configured where required

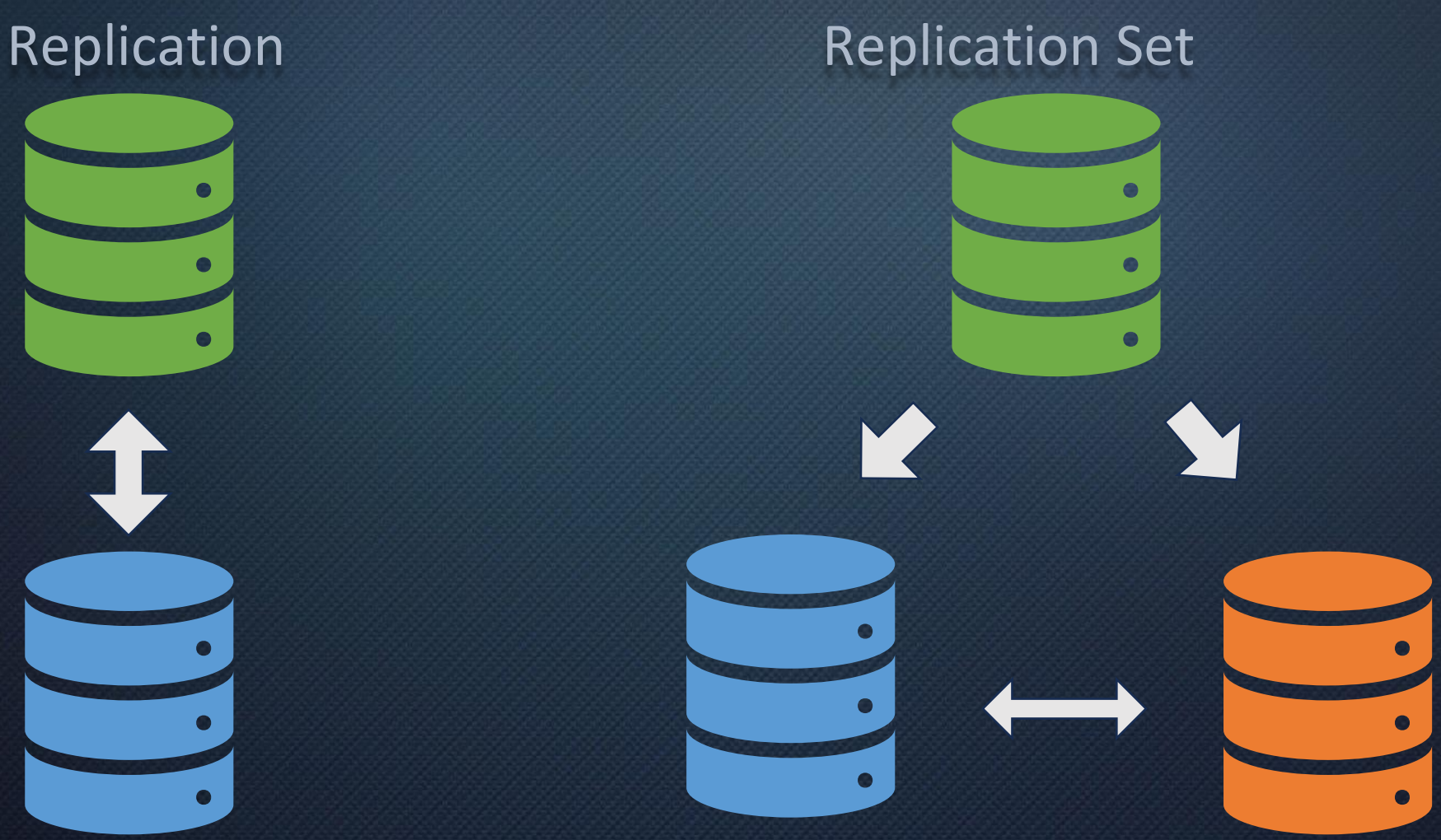
- Transparent Data Encryption
- Database Auditing
- Database Hardening

- CDC

MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Replication Types

PUG CHALLENGE



Replication Types

Server 1

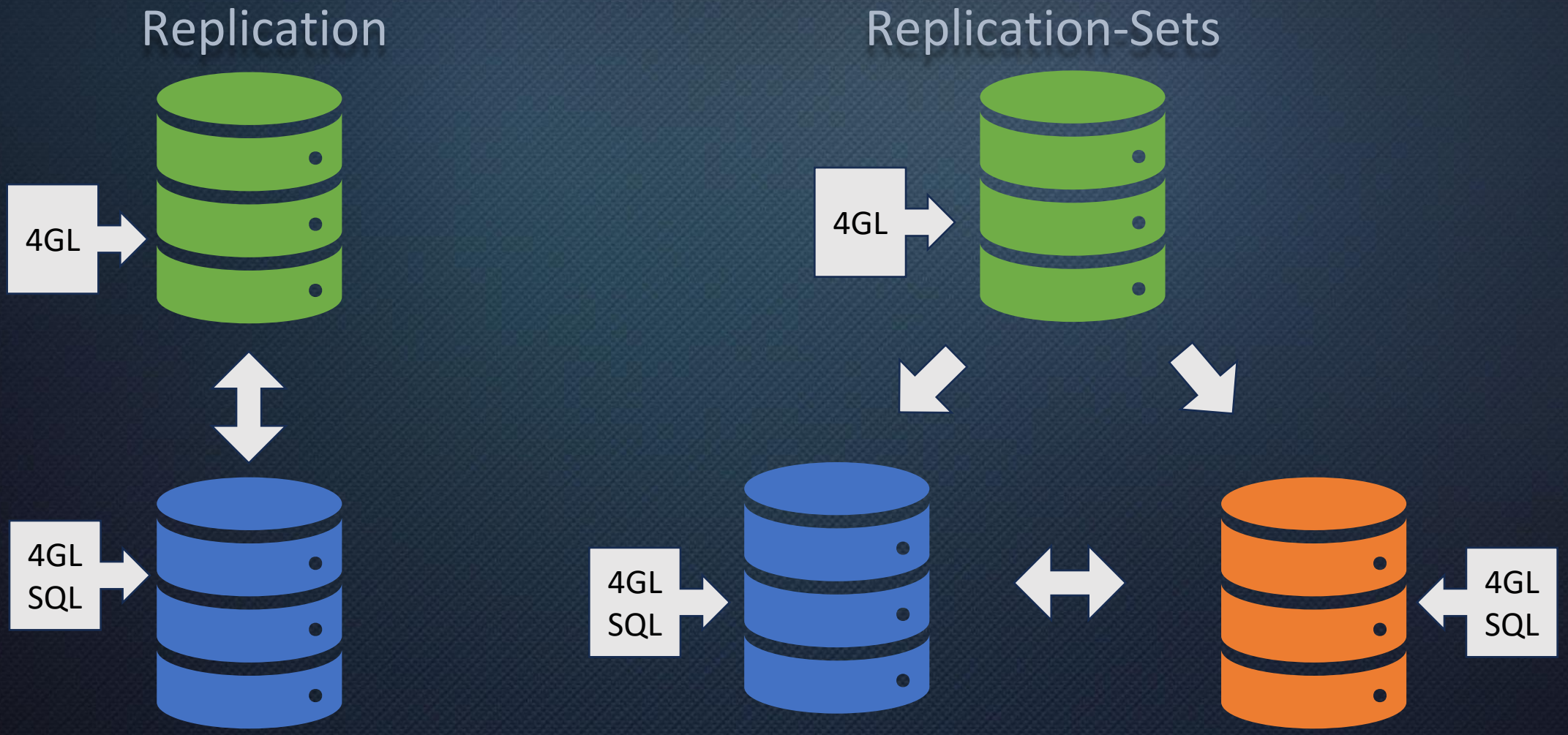
Server 2

Server 3

Server considerations

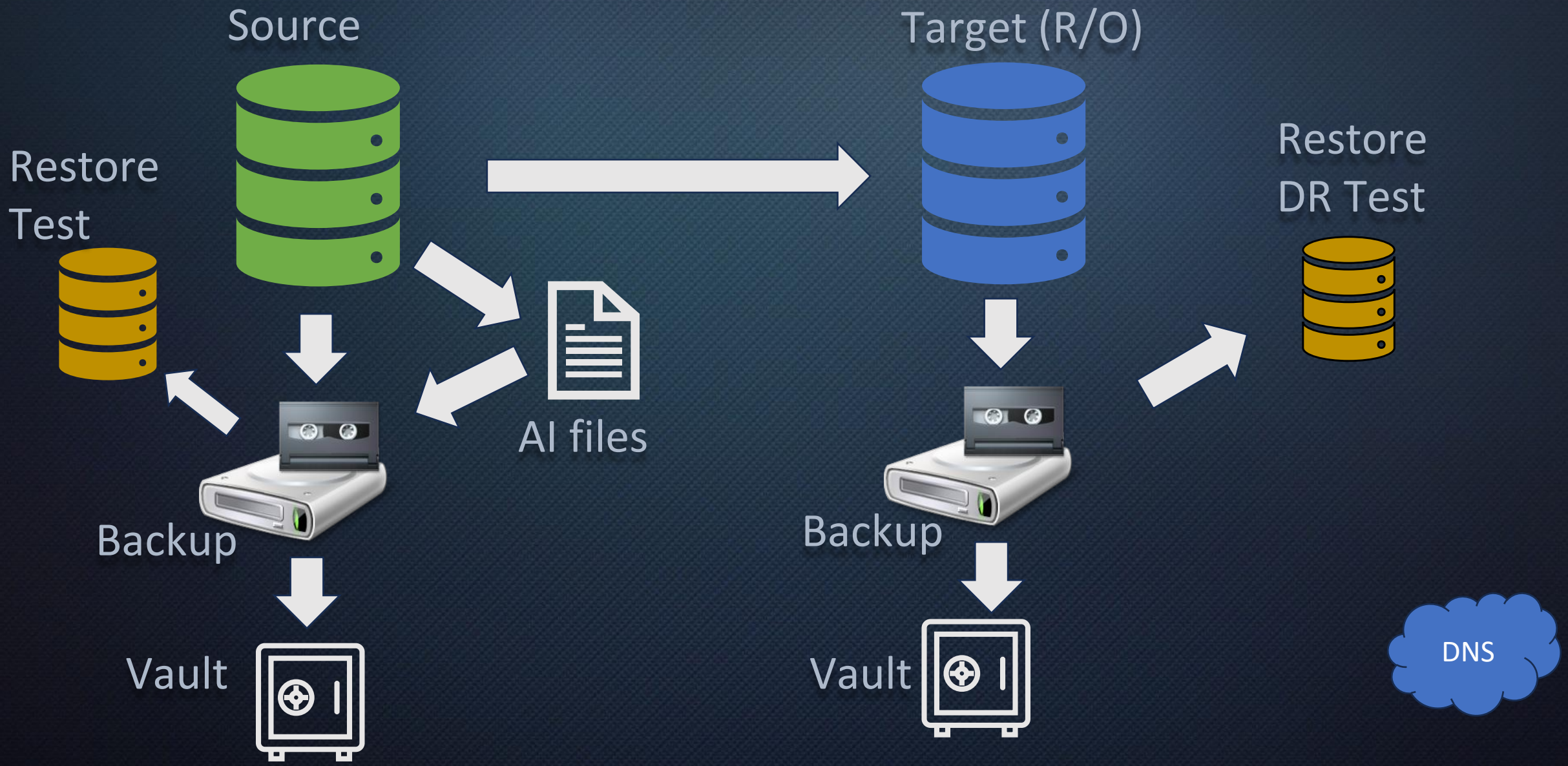
- Memory
- CPU
- Disks
- Network

Other Benefits



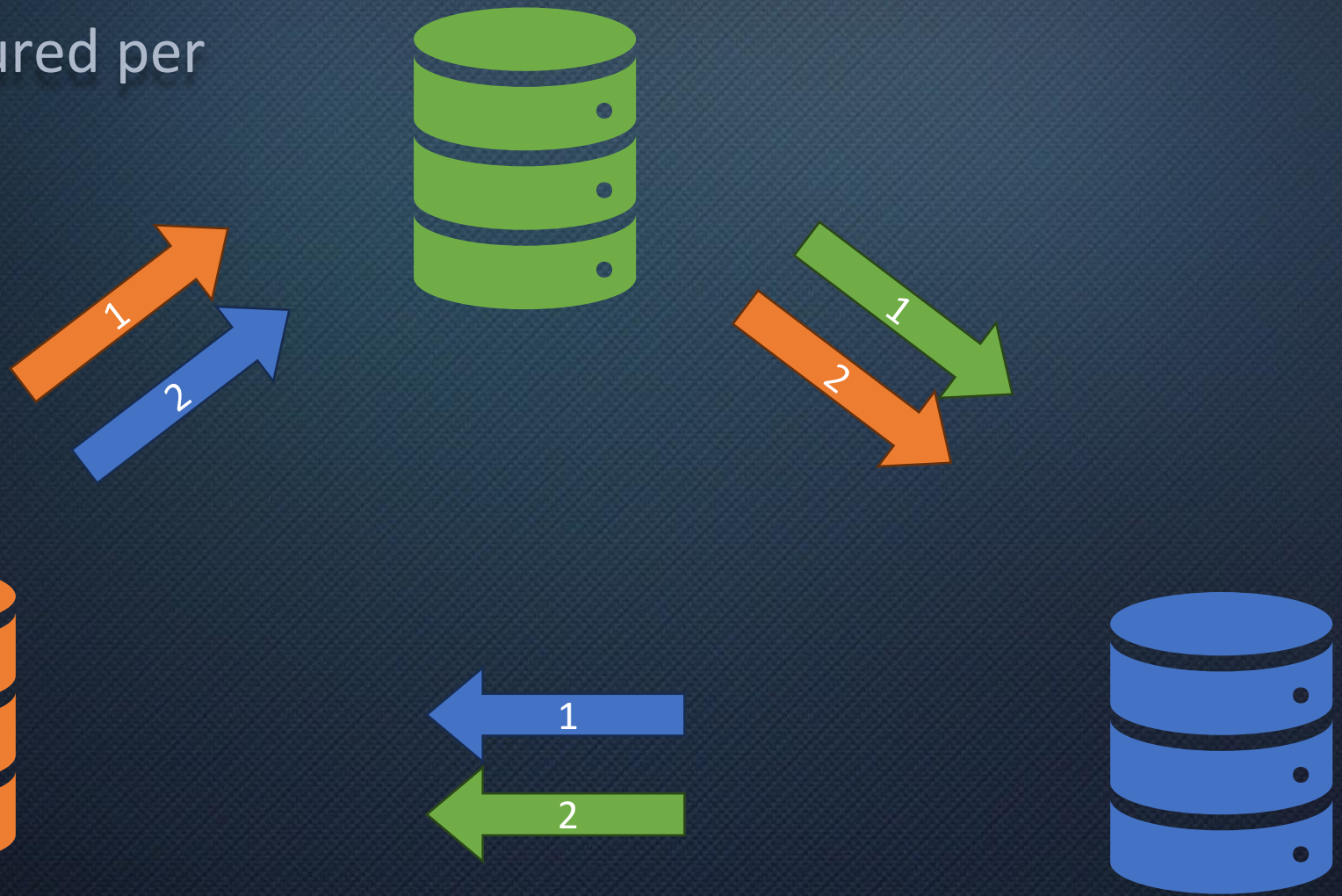
Distribute the data access across resources

Other Benefits



Replication Sets

Sets are configured per database



Replication Sets

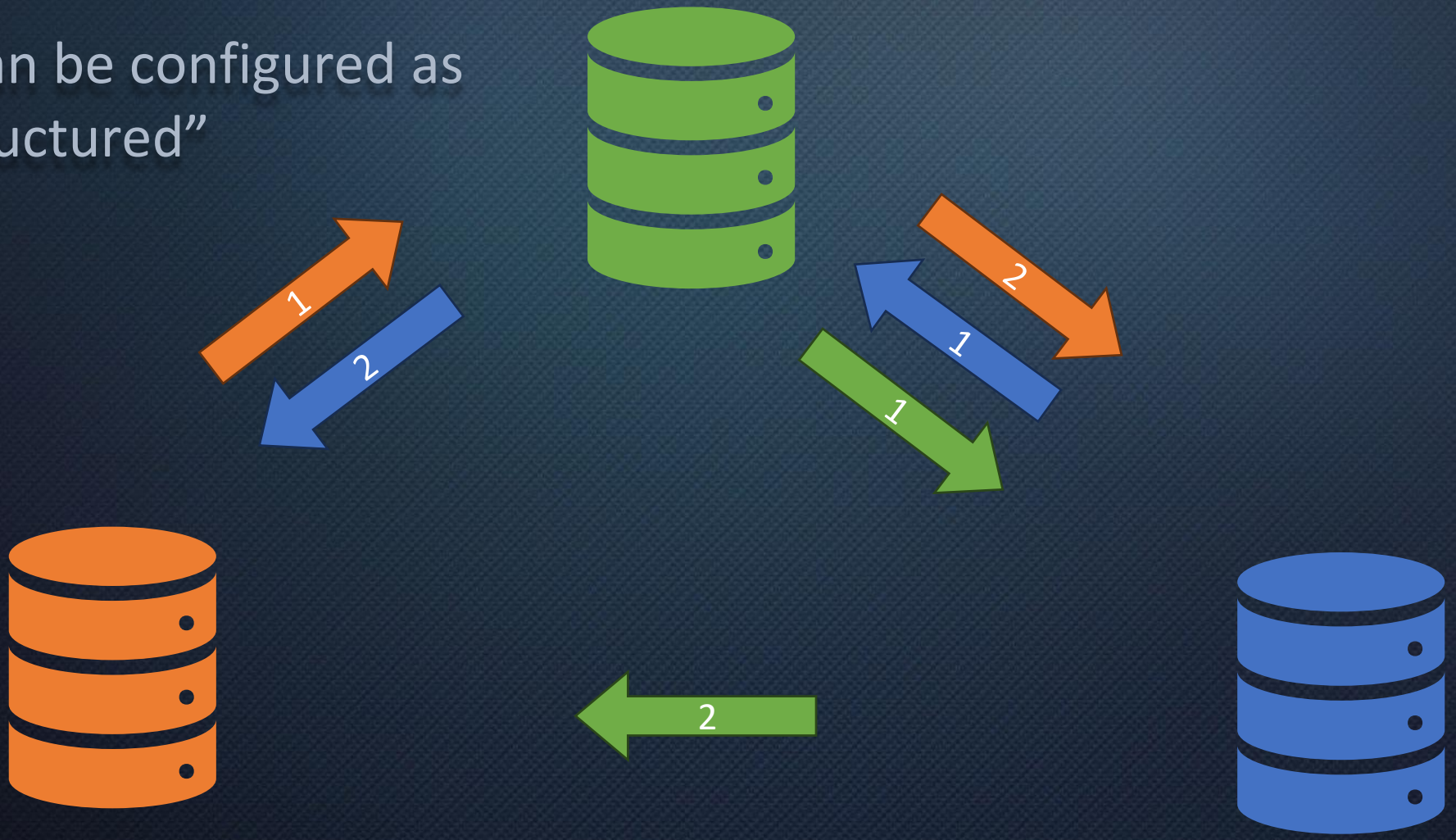
Replication Properties File : database.repl.properties

```
[server]
    control-agents=BLUE,ORANGE
    database=GREEN
    transition=manual

[control-agent.BLUE]
    name=BLUE
    database=BLUE

[control-agent.ORANGE]
    name=ORANGE
    database=ORANGE
```

```
[transition]
    replication-set=1
    transition-to-agents=BLUE,ORANGE
    database-role=reverse
    auto-begin-ai=1
    auto-add-ai-areas=0
    restart-after-transition=1
    backup-method=mark
    source-startup-arguments=-pf source.pf
    target-startup-arguments=-pf target.pf
```

Replication Sets

Sets can be configured as “unstructured”



MIP
A SOCIAL SOLUTION IN A DIGITAL DIMENSION

Agenda

PUG CHALLENGE

- Overview
- Introducing MIP
- High Availability
  - Disaster Recovery
  - Load balancing
  - Backups
  - After-Image
  - Replication
  - Replication-Sets
  - Alternate Database Connections
  - Considerations

DB Alternative connections

If a database is offline or otherwise unavailable, client-server connections can automatically connect to an alternate database, such as a replication target or backup database.

The alternate connections can only be specified when using a client-server connection, not a single-user or self-service (shared-memory) connection.

Additional startup parameters to the connect.pf



DB Alternative connections

- Connection Retry Attempts (-ct) - define the number of retries for a client-server database connection attempt.
- Alternate Database 1 (-dbalt1) - specify first alternate DB
- Alternate Database 2 (-dbalt2) - specify second alternate DB
- Retry Connect Set (-retryConnect) - number of times to retry connecting to a set of (primary and alternates), DB's before giving up
- Retry Connect Set Pause (-retryConnectPause) - how long to wait (in seconds) before attempting to reconnect

(continue)

DB Alternative connections

- The Auto Reconnect (-autoReconnect) client-session startup parameter serves two purposes:
  - For GUI and character-mode ABL sessions, indicates you want to re-establish connections to disconnected databases that were specified using startup parameters.
  - If OpenEdge replication is used, indicates that the primary replication target database should be included in the alternate database connection set, if the connection to the replication source database fails.



DB Alternative connections

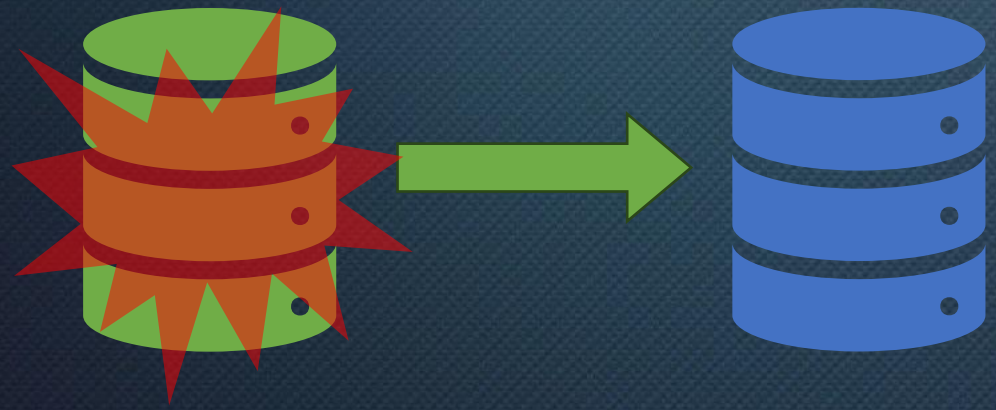
Client connection parameter file: connect.pf

- -db BLUE -H server1 -S 25511
- -ct 1
- -dbalt1 “YELLOW -H server2 -S 25512"
- -dbalt2 “ORANGE -H server3 -S 25513"
- -retryConnect 3
- -retryConnectPause 2
- -autoReconnect
- -logentrytypes DB.Connects

Failures

If the Source database “crashes”

- ✅ Replication will transition to Target database



- ✅ Client-Server connection will connect to next configured alternative database

!!! HIGH-AVAILABILITY seems possible !!!

Failures

If the Source database “crashes”

✅ Replication will transition to Target database

⁉️ Client-Server connection will connect to next configured alternative database

Failures

If the Source database is “failover” to Target

✅ Replication will transition Source to Target database

⁉ Client-Server connection will re-connect to the primary database

⁉ Client-Server connection will connect to next configured alternative database

Failures

If the Source database is “failover” to Target

✅ Replication will transition Source to Target database

⁉️ Client-Server connection is not affected at all

Considerations

- Replication Sets and Alternative Database Connection are independent

- Client-Server connections with Alternative Database Connection DOES NOT follow the SOURCE database.

- Replication databases during failover restarts with the correct Source.pf and Target.pf

- Client-server connection always connect to the primary database
- It will only connect to the alternative databases, if the primary goes ‘down’
- If the primary is down, any ‘new’ client-server connections will fail.
- You will need to update the connect.pf



Considerations

- If you want to use these options to accomplish High Availability, you will need to try and align
  - Databases connection/transition sequence
  - Timeouts and retries

- During a Failure, both ‘should’ move onto the next active online database

- During a controlled transition, rather ensure client-server connections :
  - Exit session
  - Updated to use the next connect.pf
  - Restart session



Considerations

All Client-Server (and Shared memory) connections must be evaluated and aligned

- PAS
- GUI
- CHUI

- SQL

- CDC
  - Shared memory vs Client-server
  - Should be managed in line with Replication and follow swith-over accordingly

- Other
  - REST
  - API

Conclusion

To achieve High Availability and Connecting and access to the Correct, Active and Updatable database comes with its own challenges.

!!! REPLICATION, REPLICATION, REPLICATION !!!

Understand and manage your application connections.

One step closer to achieving your High Availability goals …

OVERVIEW

Q&A