

HA-OSCAR



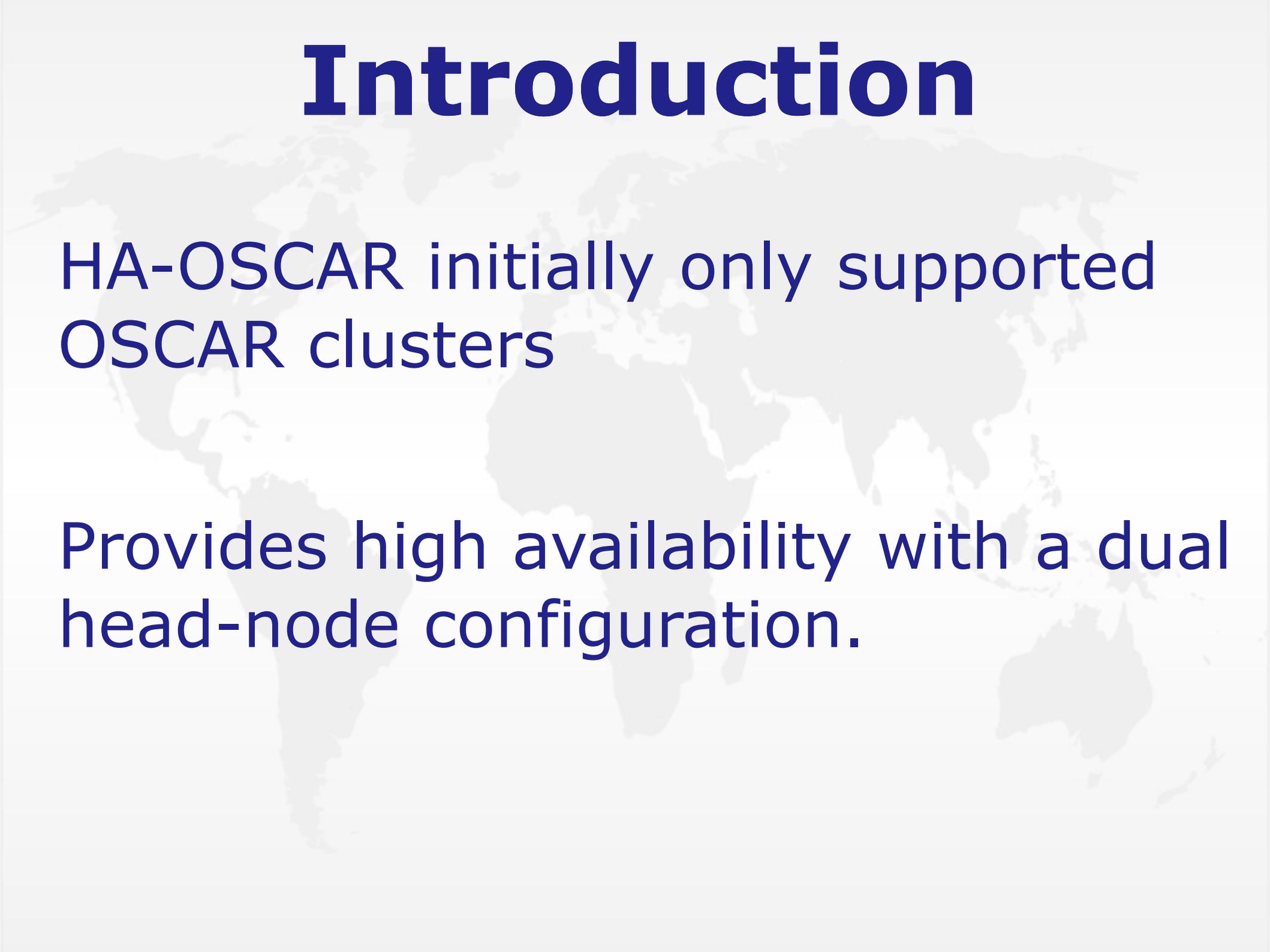
Chuka Okoye
Himanshu Chhetri



What is HA-OSCAR?

**“High Availability Open Source Cluster
Application Resources”**

Introduction

A faint, light-colored world map is visible in the background of the slide, centered behind the text.

HA-OSCAR initially only supported OSCAR clusters

Provides high availability with a dual head-node configuration.

HA-OSCAR 2.0

Independent of OSCAR since 1.3+

Completely re-architected to be very modular and extensible.

Currently runs on 15 flavors of Linux

Project Management

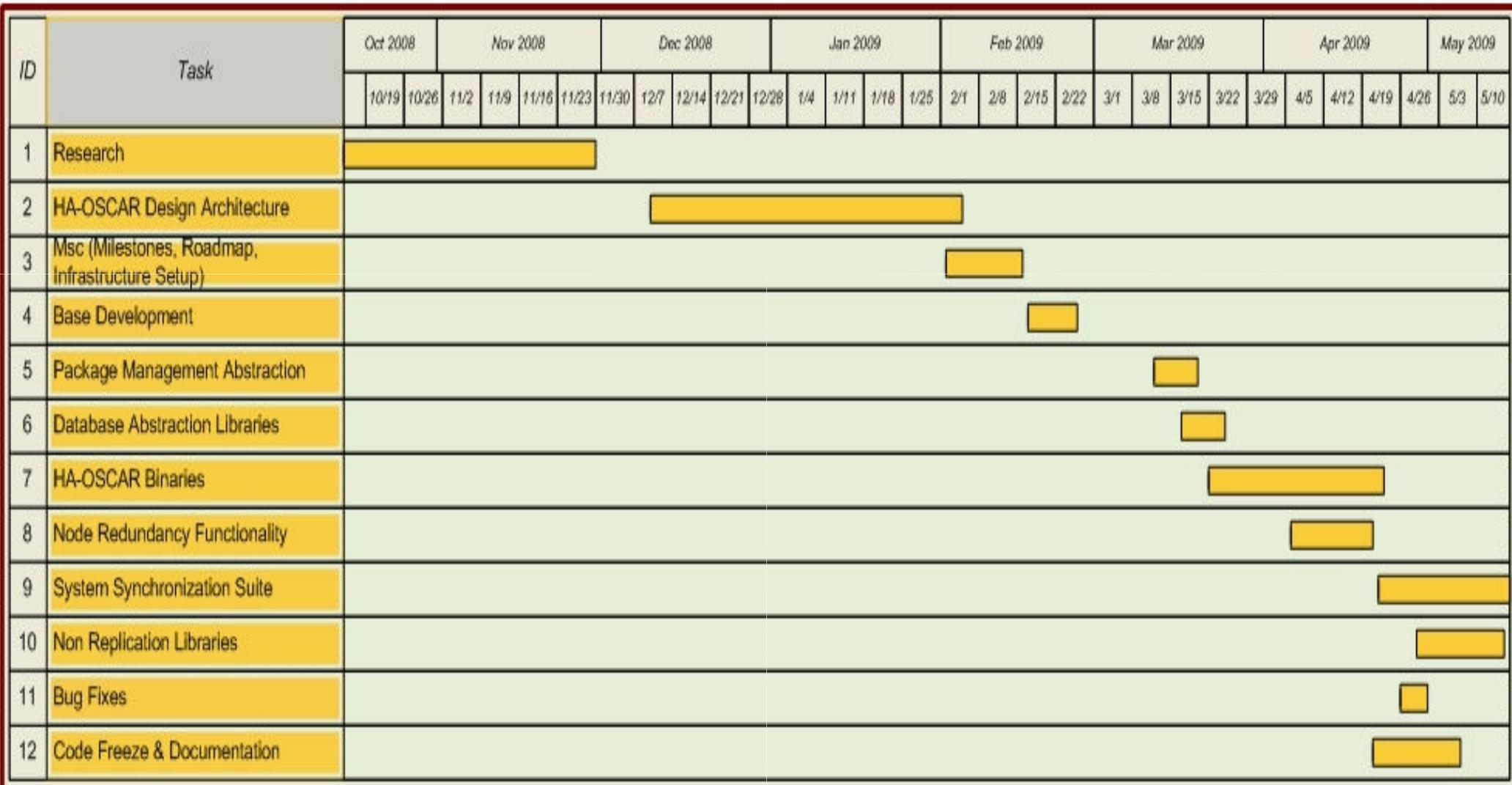


Code hosted on Google Code which provides “Issues” tracker and Subversion based collaborative source code hosting.

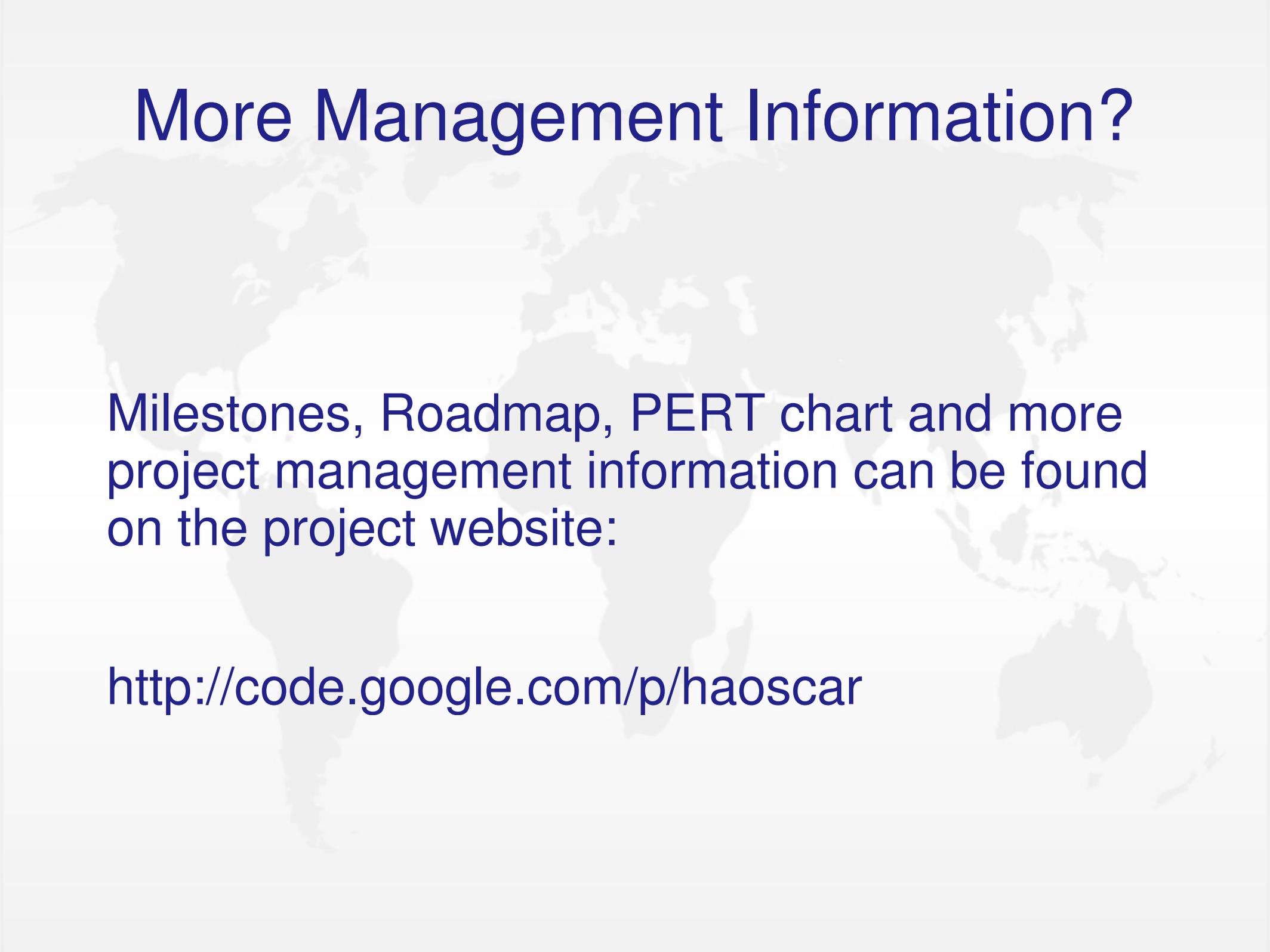
Cluster of four different Linux workstations used for development and testing.

Microsoft Office Live Space for more coordinated team collaboration.

Gantt Chart



More Management Information?



Milestones, Roadmap, PERT chart and more project management information can be found on the project website:

<http://code.google.com/p/haoscar>

Architecture

```
graph TD; A[HA-OSCAR Binaries] --- B[Core High Availability Installation Framework]; A --- C[High Availability Tools Configuration]; A --- D[Web Services Framework]; A --- E[HA-OSCAR API];
```

HA-OSCAR
Binaries

Core High Availability
Installation Framework

High Availability Tools
Configuration

Web Services Framework

HA-OSCAR API

CHAIIF Architecture

Database
Abstraction Library

System Profiling
Services

Package
Management

CHAIF

Core **H**igh **A**vailability **I**nstallation **F**ramework

Foundation of an HA-OSCAR installation procedure

Provides common services to other frameworks:

- Package management abstraction
- System profiling services
- Database drivers
- Environmental sanity checks

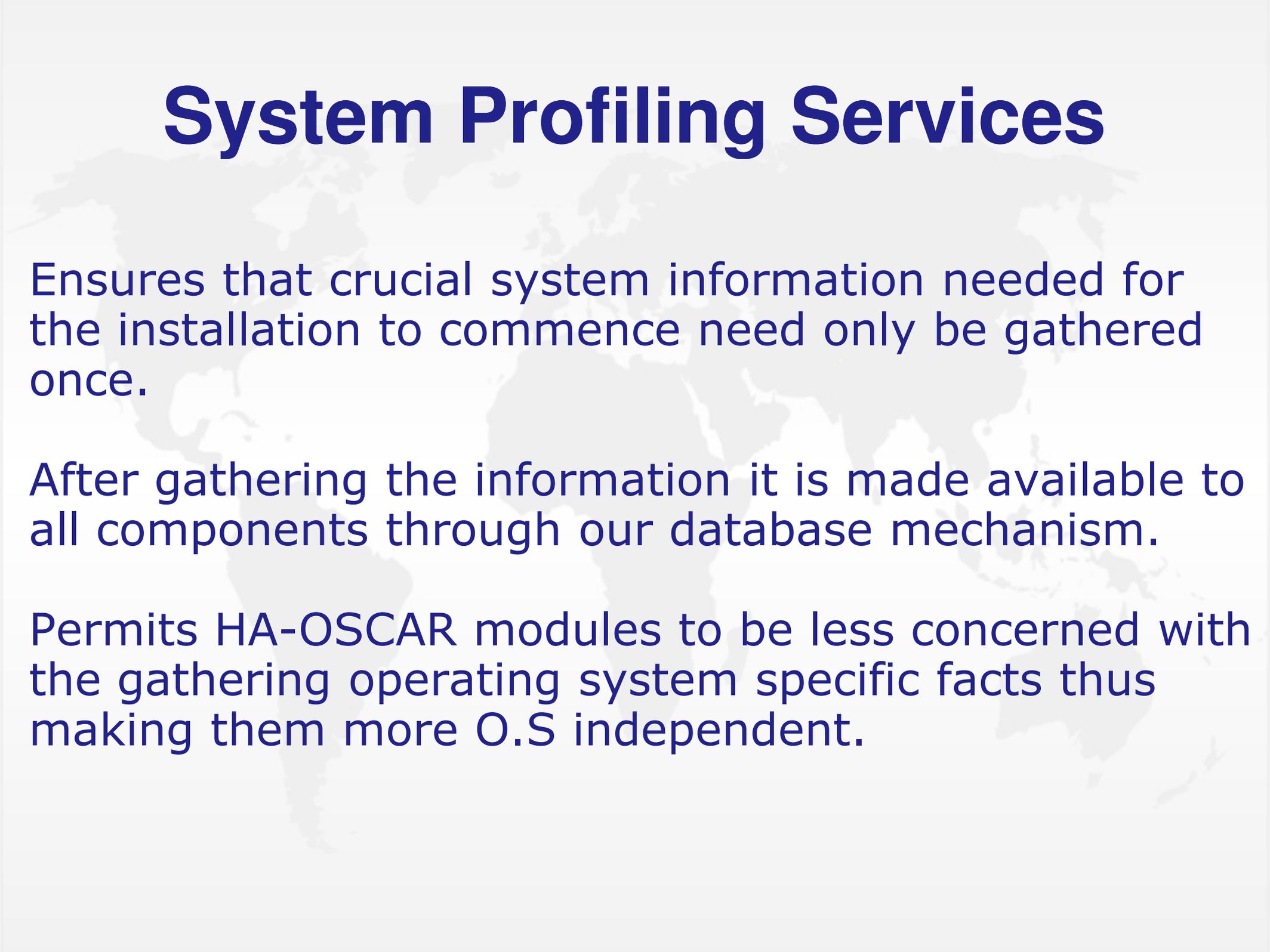
Package Management

Makes it convenient for user to install HA-OSCAR using standard Linux package manager like apt and yum

New addition to HA-OSCAR. It allows platform independent code to be written without any worries of differences in services offered by the native package management

Although, the package management abstraction is still in its infancy, it currently supports fifteen (15) Linux operating systems

System Profiling Services

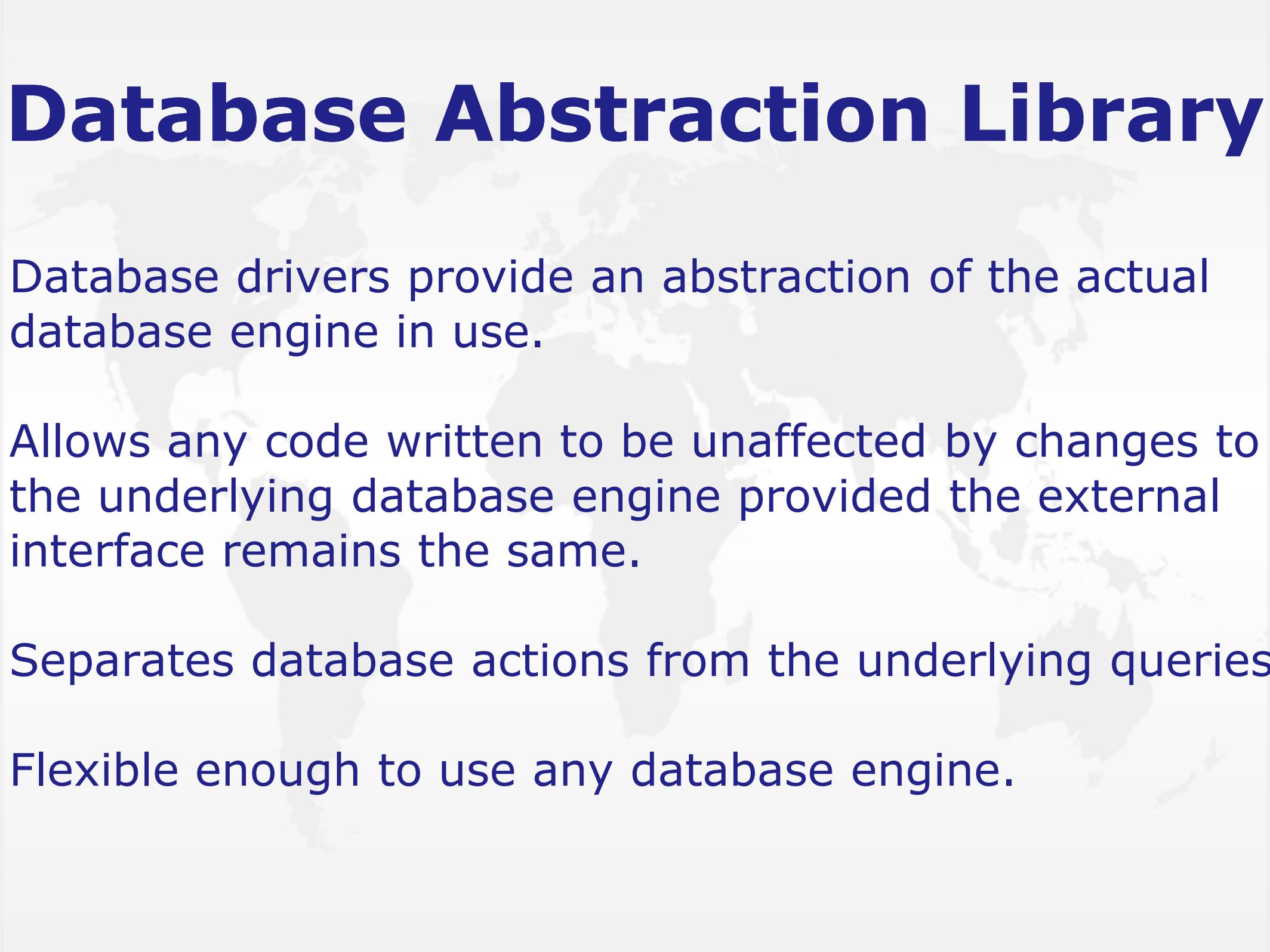


Ensures that crucial system information needed for the installation to commence need only be gathered once.

After gathering the information it is made available to all components through our database mechanism.

Permits HA-OSCAR modules to be less concerned with the gathering operating system specific facts thus making them more O.S independent.

Database Abstraction Library



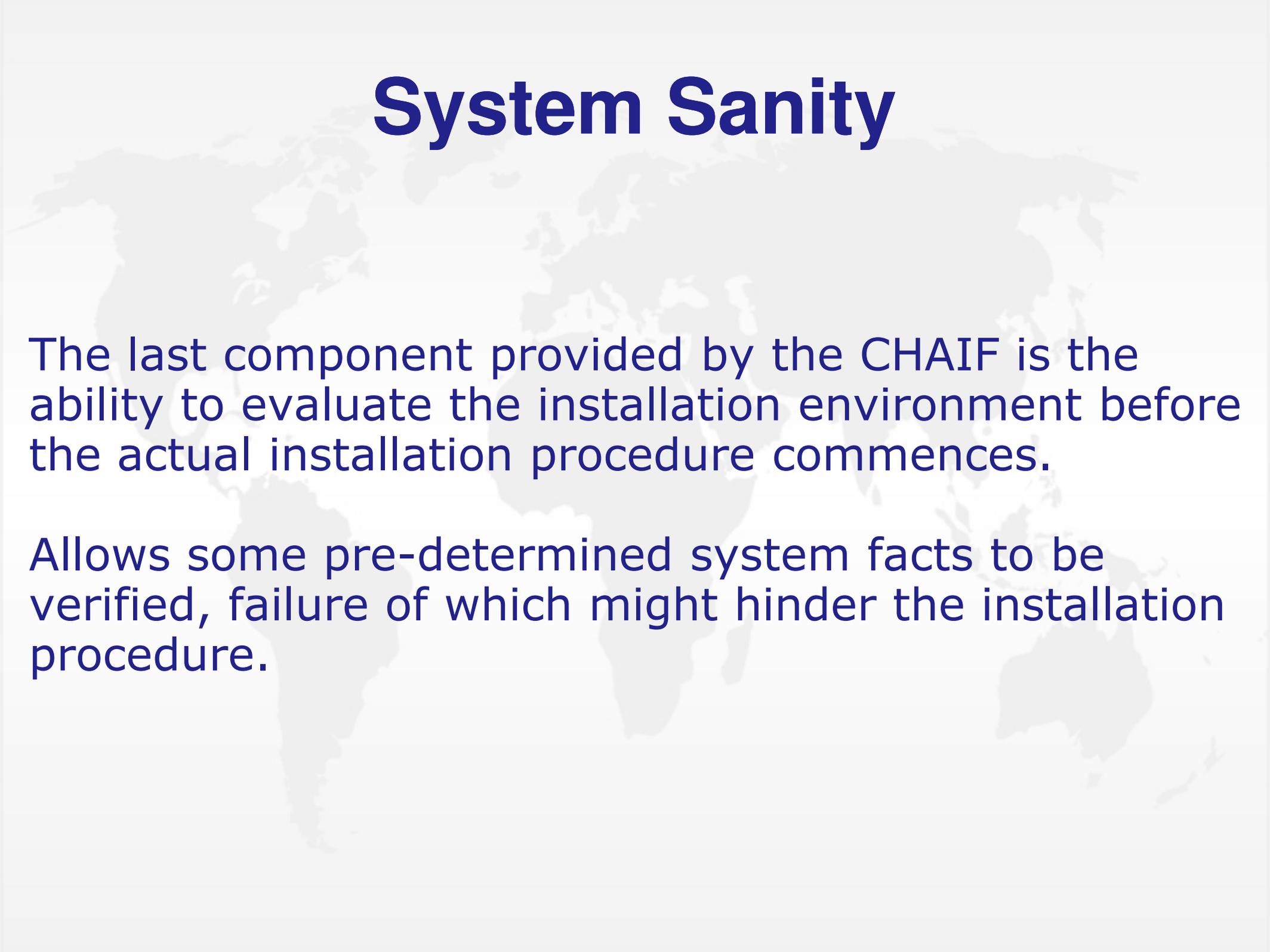
Database drivers provide an abstraction of the actual database engine in use.

Allows any code written to be unaffected by changes to the underlying database engine provided the external interface remains the same.

Separates database actions from the underlying queries

Flexible enough to use any database engine.

System Sanity

A faint, light gray world map is visible in the background of the slide, centered behind the text.

The last component provided by the CHAIF is the ability to evaluate the installation environment before the actual installation procedure commences.

Allows some pre-determined system facts to be verified, failure of which might hinder the installation procedure.

HATCI

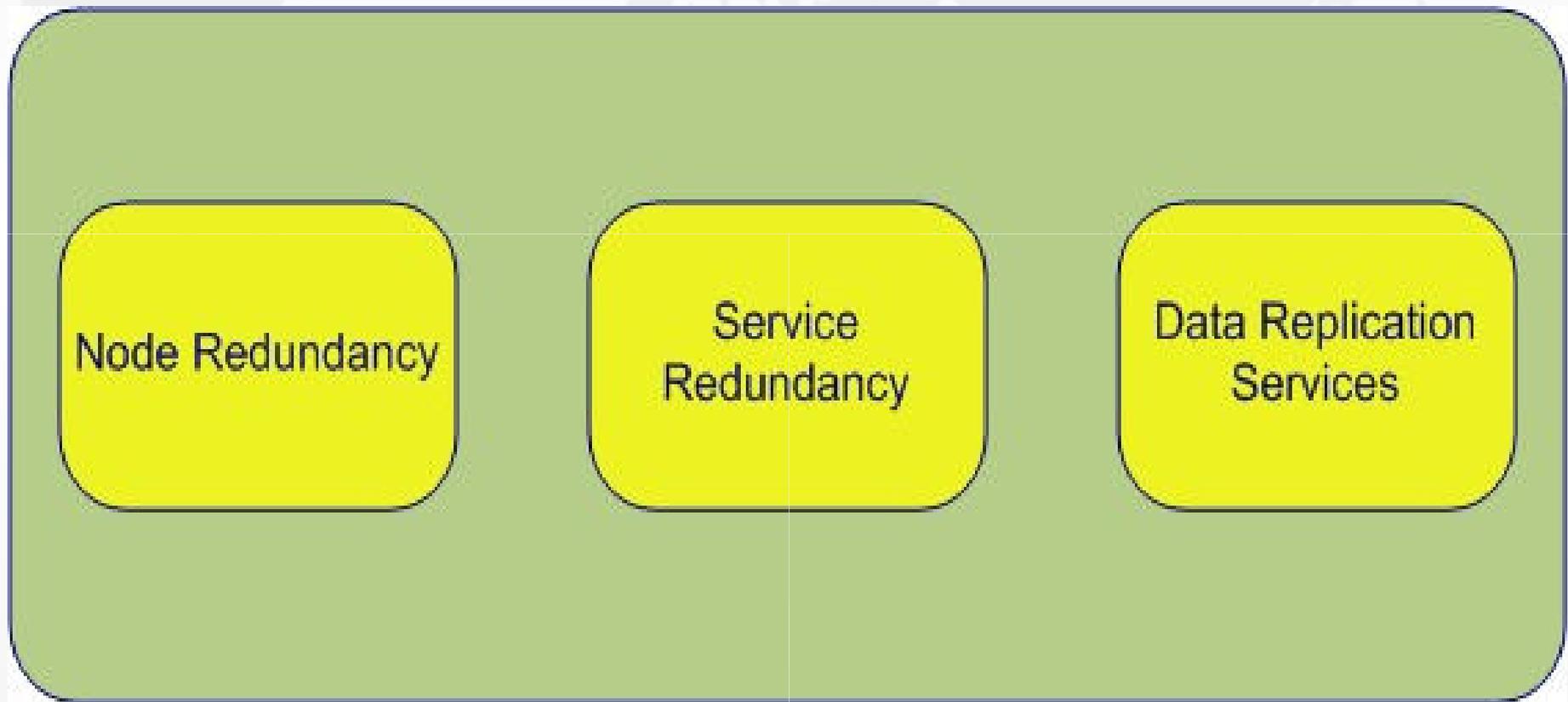
High Availability Tools Configuration and Installation

Responsible for the installation and configuration of tools necessary to provide high availability to the system

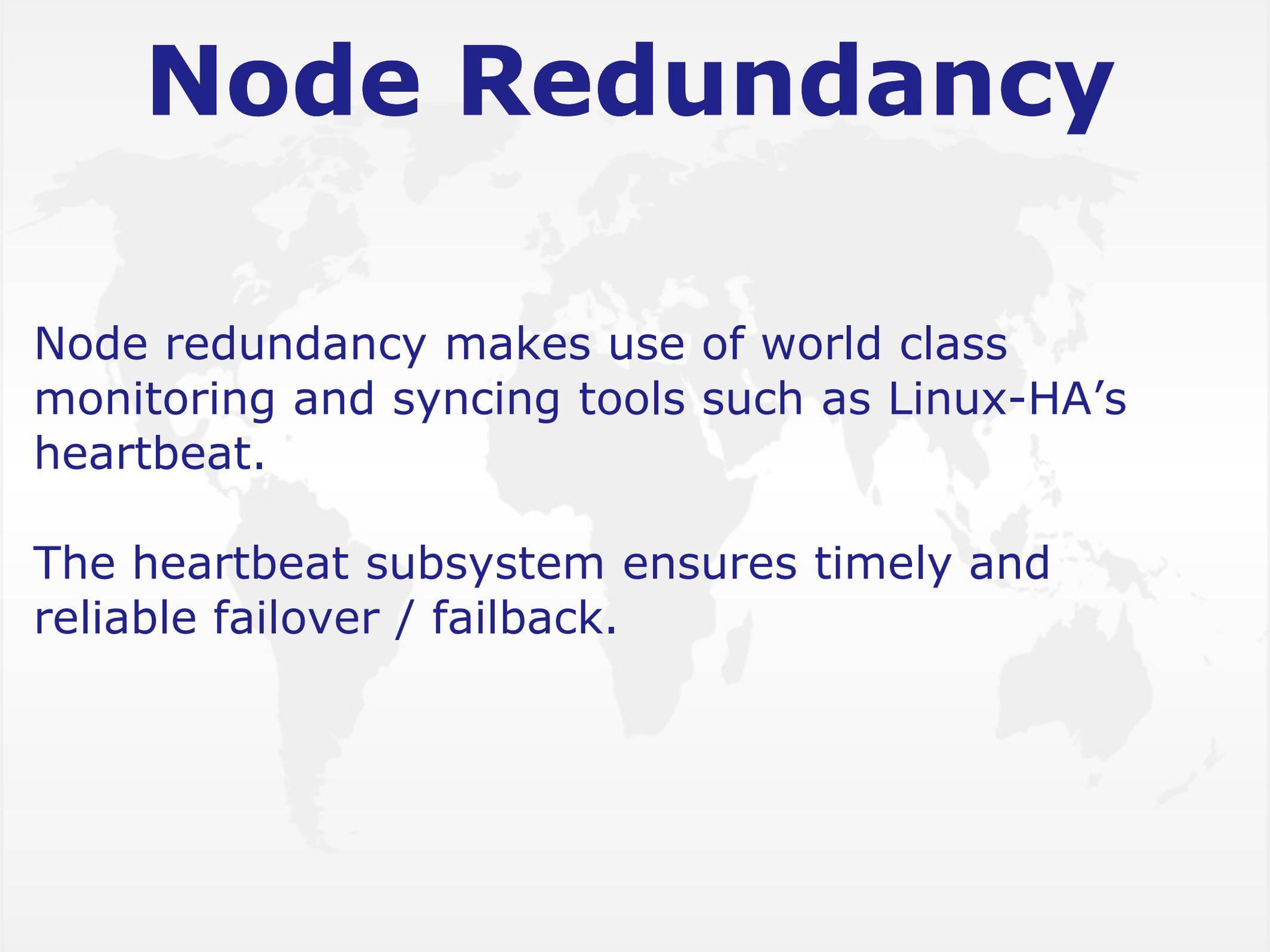
The current implementation of HATCI is divided into three sub components:

- Node Redundancy
- Service Redundancy
- Data Replication Services

HATCI Architecture



Node Redundancy



Node redundancy makes use of world class monitoring and syncing tools such as Linux-HA's heartbeat.

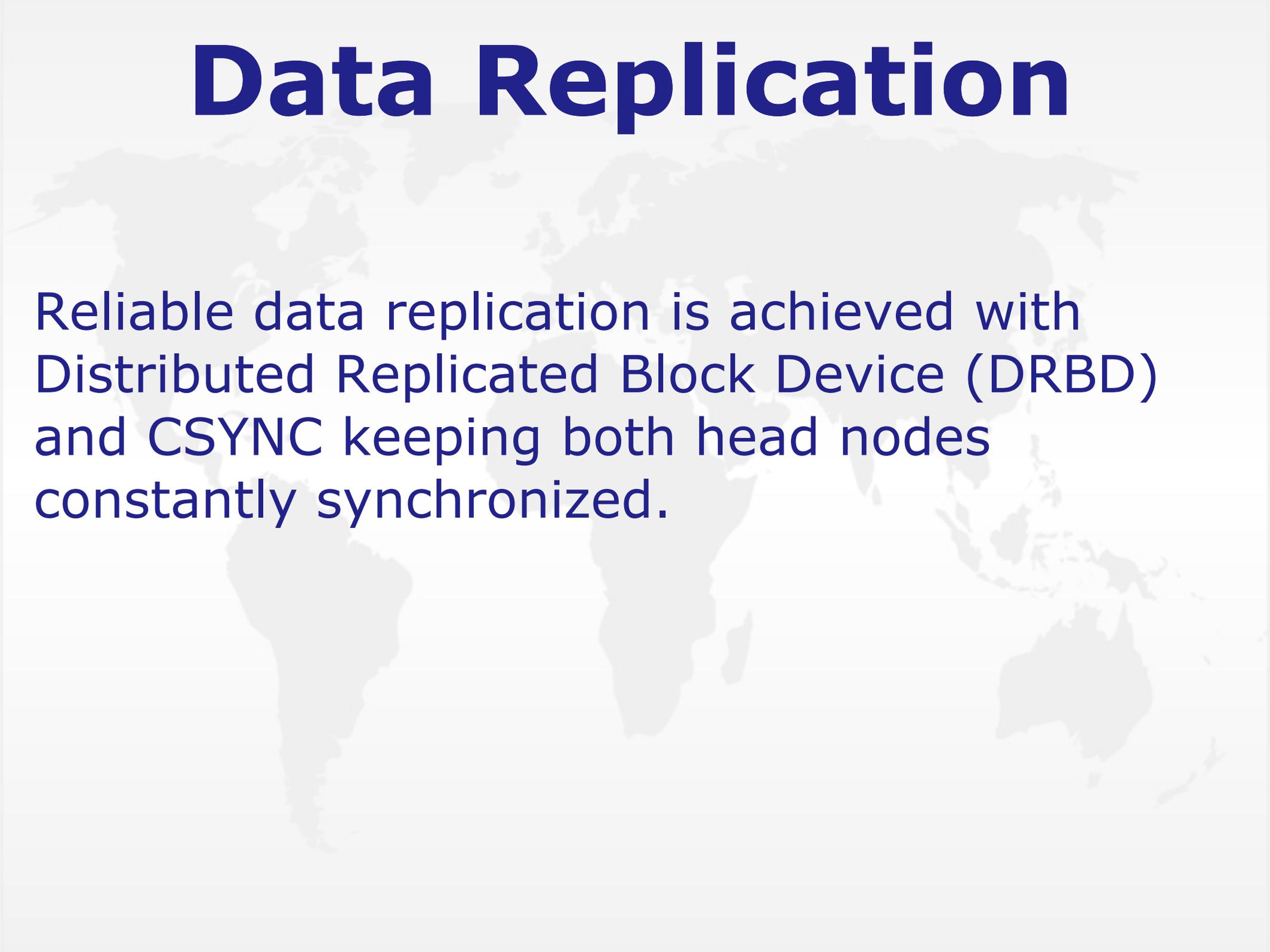
The heartbeat subsystem ensures timely and reliable failover / failback.

Service Redundancy



Service redundancy on the other hand is responsible for the configuration and installation of tools used to provide highly available cluster services.

Data Replication



Reliable data replication is achieved with Distributed Replicated Block Device (DRBD) and CSYNC keeping both head nodes constantly synchronized.

HA-OSCAR API

New addition in which developers and administrators can extend the functionality of HA-OSCAR using provided hooks

Allows creating event notification services and powerful rule based systems

Can be used to determine the state of the monitored services

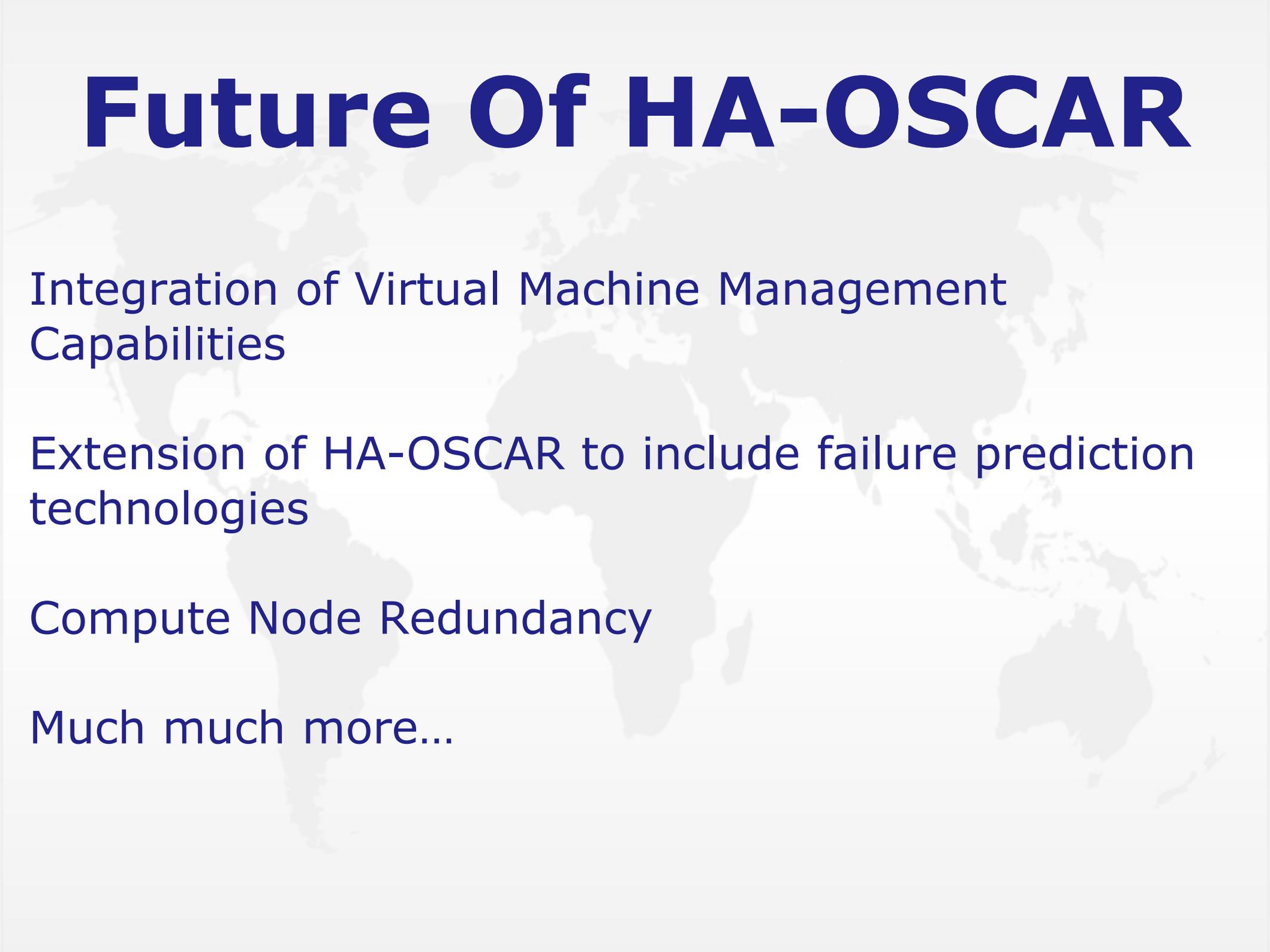
Web Services Framework



Another new addition to HA-OSCAR which allows installation and management to be conducted from a web browser.

It also provides a much needed separation of user interface code from the core HA installation procedure.

Future Of HA-OSCAR



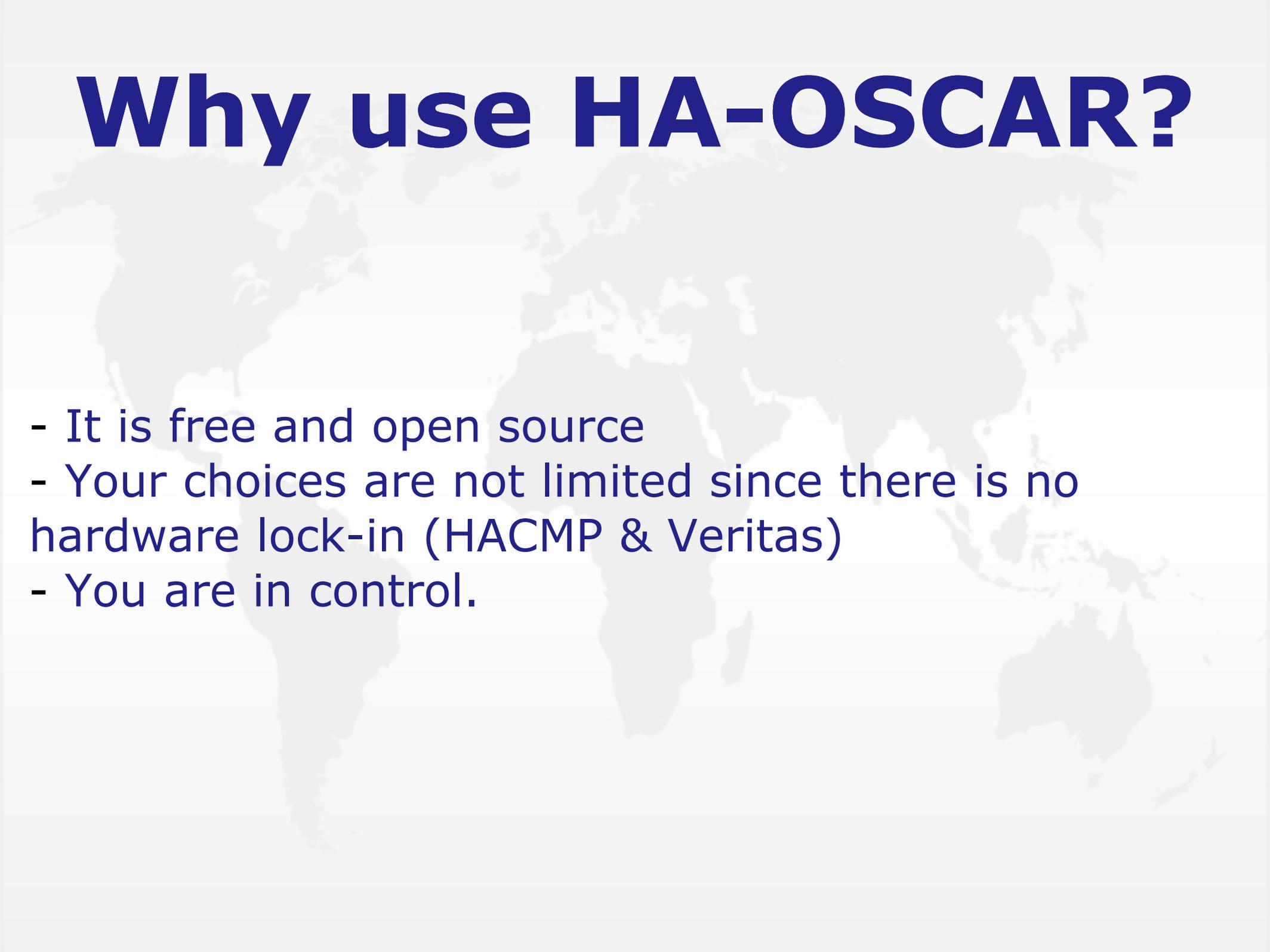
Integration of Virtual Machine Management Capabilities

Extension of HA-OSCAR to include failure prediction technologies

Compute Node Redundancy

Much much more...

Why use HA-OSCAR?



- It is free and open source
- Your choices are not limited since there is no hardware lock-in (HACMP & Veritas)
- You are in control.



Demonstration