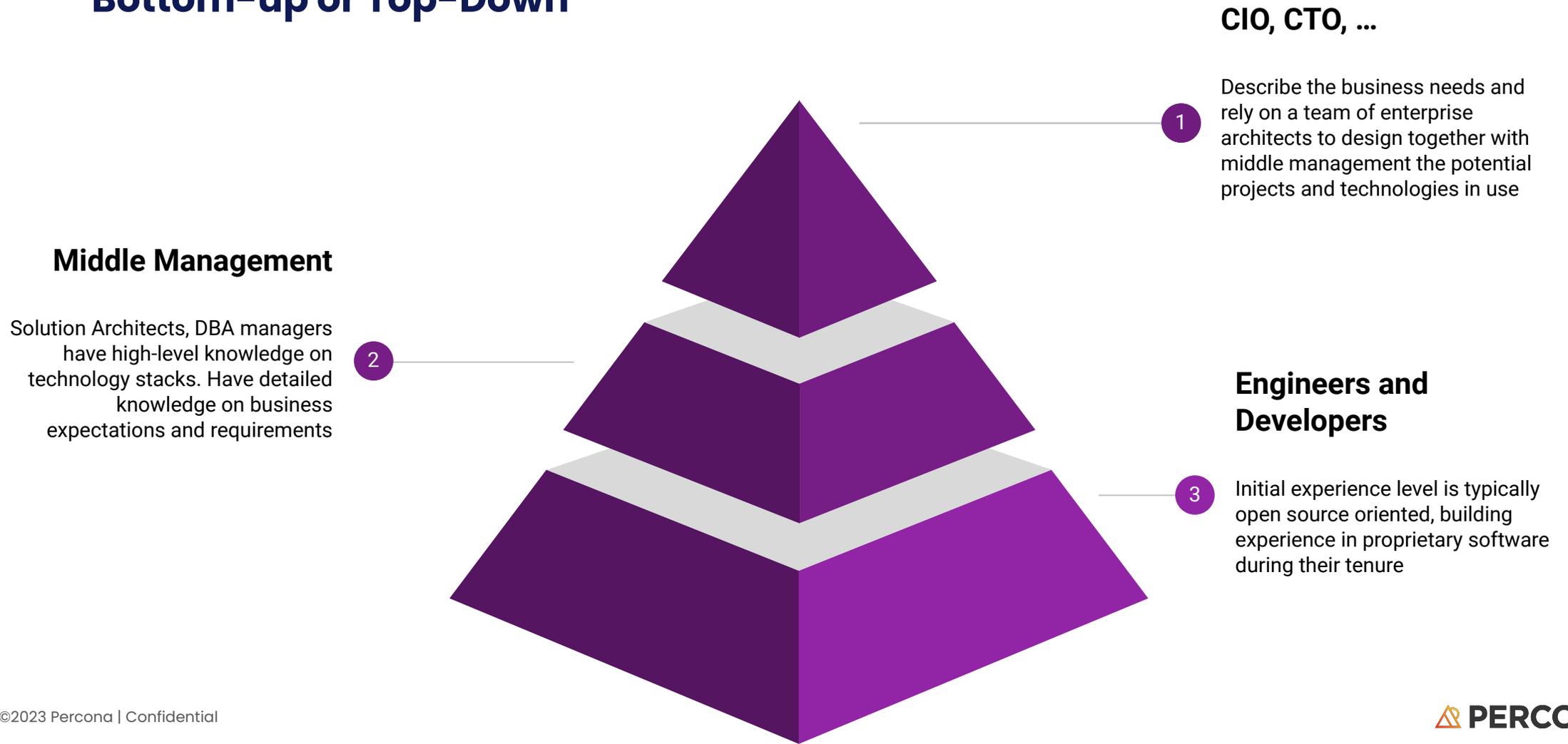# PERCONA

Databases run better with Percona

Agenda

- The move from proprietary databases to Open Source
- The benefits of moving towards Open Source adoption
- The risks associated with Open Source adoption
- Percona as a expertise partner

PERCONA

Typical scenarios

- New emerging technologies are built with Open Source solutions
- Standardisation effort inside the organization (cataloguing technology)
- Restructuring and cost efficiency efforts
- Agility effort in building scalable technologies

Landscape: Typical Open Source Adoption

Bottom-up or Top-Down

CIO, CTO, ...

1 Describe the business needs and rely on a team of enterprise architects to design together with middle management the potential projects and technologies in use

Middle Management

2 Solution Architects, DBA managers have high-level knowledge on technology stacks. Have detailed knowledge on business expectations and requirements

Engineers and Developers

3 Initial experience level is typically open source oriented, building experience in proprietary software during their tenure

PERCONA

EXPECTATION

WHY

Why?

Professional Developers

Database	Percentage
PostgreSQL	46.48%
MySQL	45.68%
SQLite	30.83%
Microsoft SQL Server	28.77%
MongoDB	28.29%
Redis	24.97%
MariaDB	17.91%
Elasticsearch	13.9%
Oracle	11.79%

Learning to code

Database	Percentage
MySQL	58.4%
MongoDB	31.32%
SQLite	30.41%
PostgreSQL	25.54%
Microsoft SQL Server	16.34%
MariaDB	14.3%
Firebase Realtime Database	11.56%
Oracle	10.84%
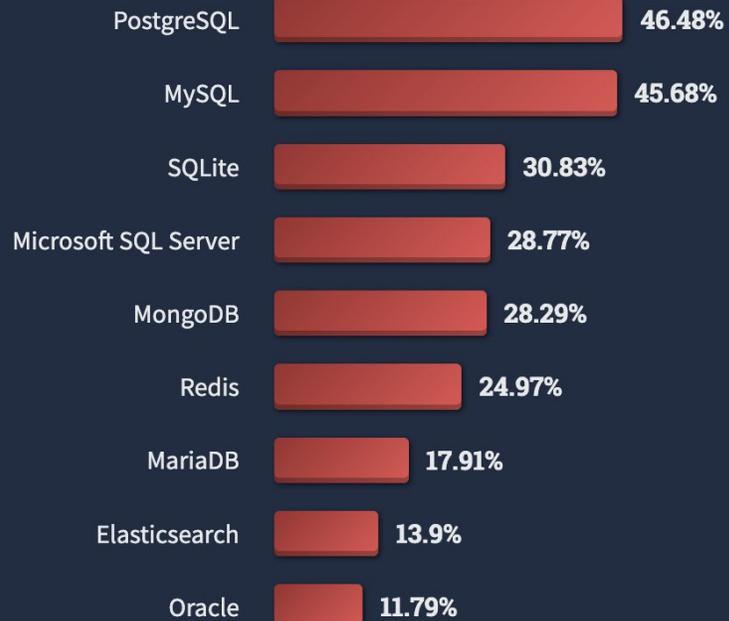
Source: Stackoverflow - Developers Survey 2022
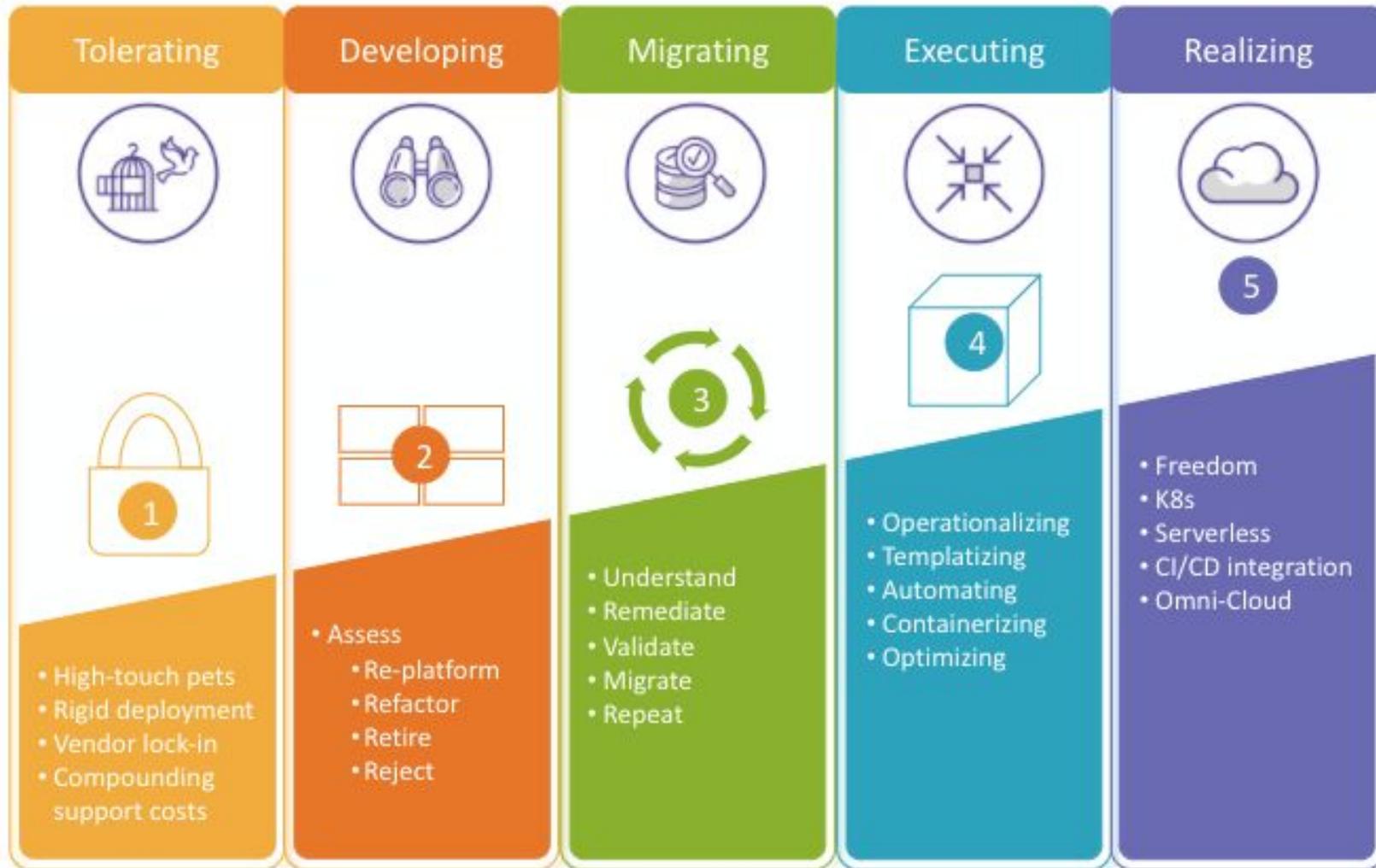
Why

- Strategic Vision
- Cloudnative
- License cost
  - Lowering cost of ownership (TCO)
  - Different concept
- Driving Adoption with adaptable platforms
- Security through mass adoption and code review
- Resilience
- Augmented service quality
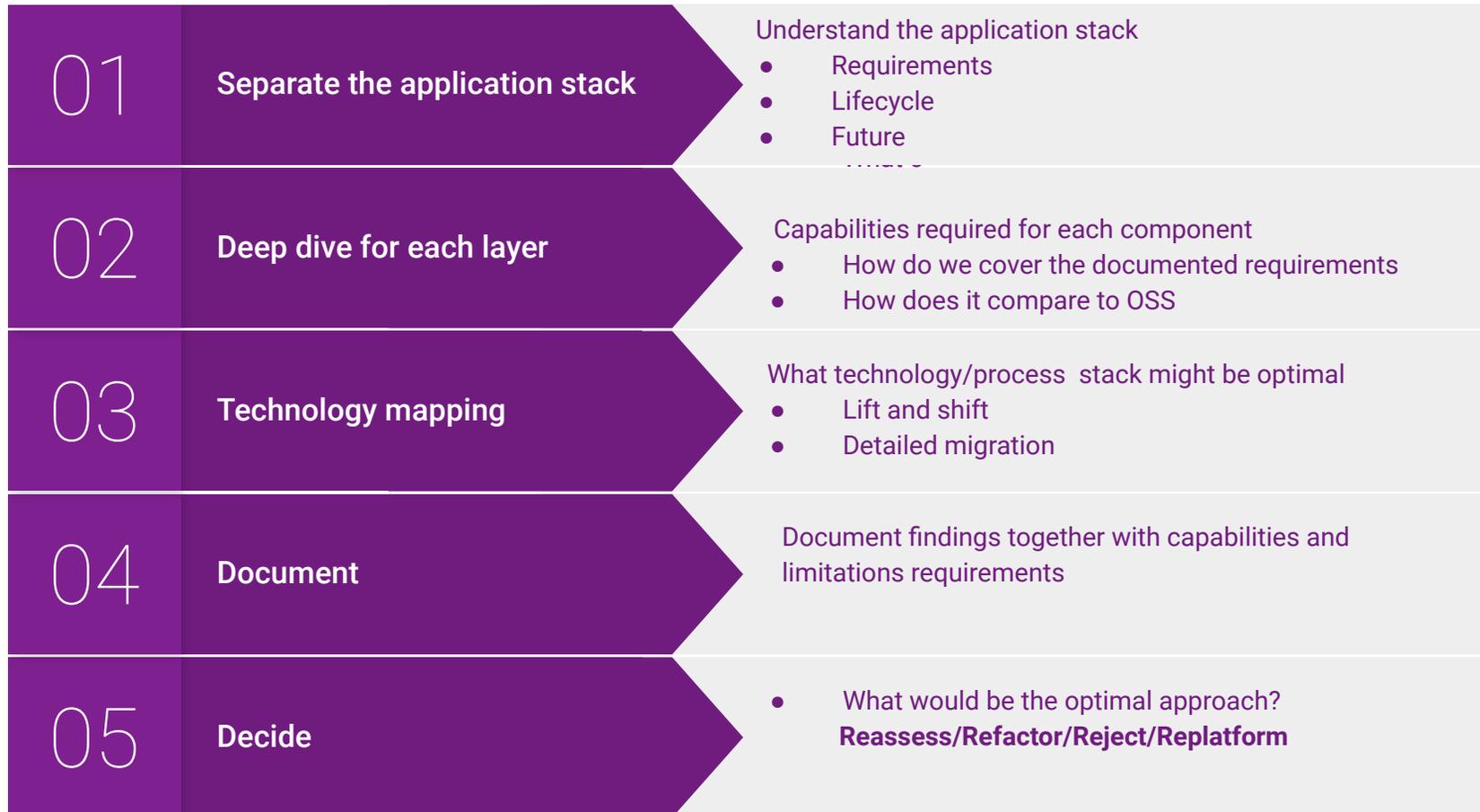- Technology Neutral
- Hiring benefits

PERCONA

Sure but How?

How can you migrate towards OSS

Tolerating	Developing	Migrating	Executing	Realizing
1	**2**	**3**	**4**	**5**
• High-touch pets	• Assess	• Understand	• Operationalizing	• Freedom
• Rigid deployment	• Re-platform	• Remediate	• Templatizing	• K8s
• Vendor lock-in	• Refactor	• Validate	• Automating	• Serverless
• Compounding support costs	• Retire	• Migrate	• Containerizing	• CI/CD integration
	• Reject	• Repeat	• Optimizing	• Omni-Cloud

Developing Your Plan

01	**Separate the application stack**	Understand the application stack • Requirements • Lifecycle • Future
02	**Deep dive for each layer**	Capabilities required for each component • How do we cover the documented requirements • How does it compare to OSS
03	**Technology mapping**	What technology/process stack might be optimal • Lift and shift • Detailed migration
04	**Document**	Document findings together with capabilities and limitations requirements
05	**Decide**	• What would be the optimal approach? **Reassess/Refactor/Reject/Replatform**

Timeline example

Commence assessing
other applications

Start

Analysis

POC

Deployment

Strategy decision

Analysis and Assessment

POC of selected applications

Deployment of Selected Applications

Example intermediary Landscape

GOLD tier

Technology Stack

- Databases
  - Oracle DB
  - Microsoft SQL
  - DB2

- Application
  - Websphere
  - JBoss
- Security
  - Webseal
  - TIFM

Silver Tier

Technology Stack

- Databases
  - Oracle DB
  - PostgreSQL
  - MySQL
  - MongoDB
- Application
  - Websphere
  - JBoss
  - Tomcat
  - Django
- Security
  - OpenAM
  - …

Bronze Tier

Technology Stack

- Databases
  - PostgreSQL
  - MySQL
  - MongoDB
- Application
  - Tomcat
  - Django
- Security
  - OpenAM
  - …

Example Target Landscape

GOLD tier

Technology Stack

- Databases
  - Oracle DB
  - PostgreSQL
  - MongoDB

- Application
  - Websphere
  - JBoss
  - Tomcat
  - Django
- Security
  - OpenAM

Silver Tier

Technology Stack

- Databases
  - PostgreSQL
  - MySQL
  - MongoDB
- Application
  - Tomcat
  - Django
- Security
  - OpenAM
  - …

Bronze Tier

Technology Stack

- Databases
  - PostgreSQL
  - MySQL
  - MongoDB
- Application
  - Tomcat
  - Django
- Security
  - OpenAM
  - …

Database technologies

- Relational database models
  - Historical choices: Microsoft SQL, Oracle DB, DB2
  - Popular Open Source choices: PostgreSQL, MySQL,...
- NoSQL database models
  - Proprietary solutions: Oracle JSON database, DocumentDB, ...
  - Popular open source: MongoDB, FerretDB
- Many other types of databases
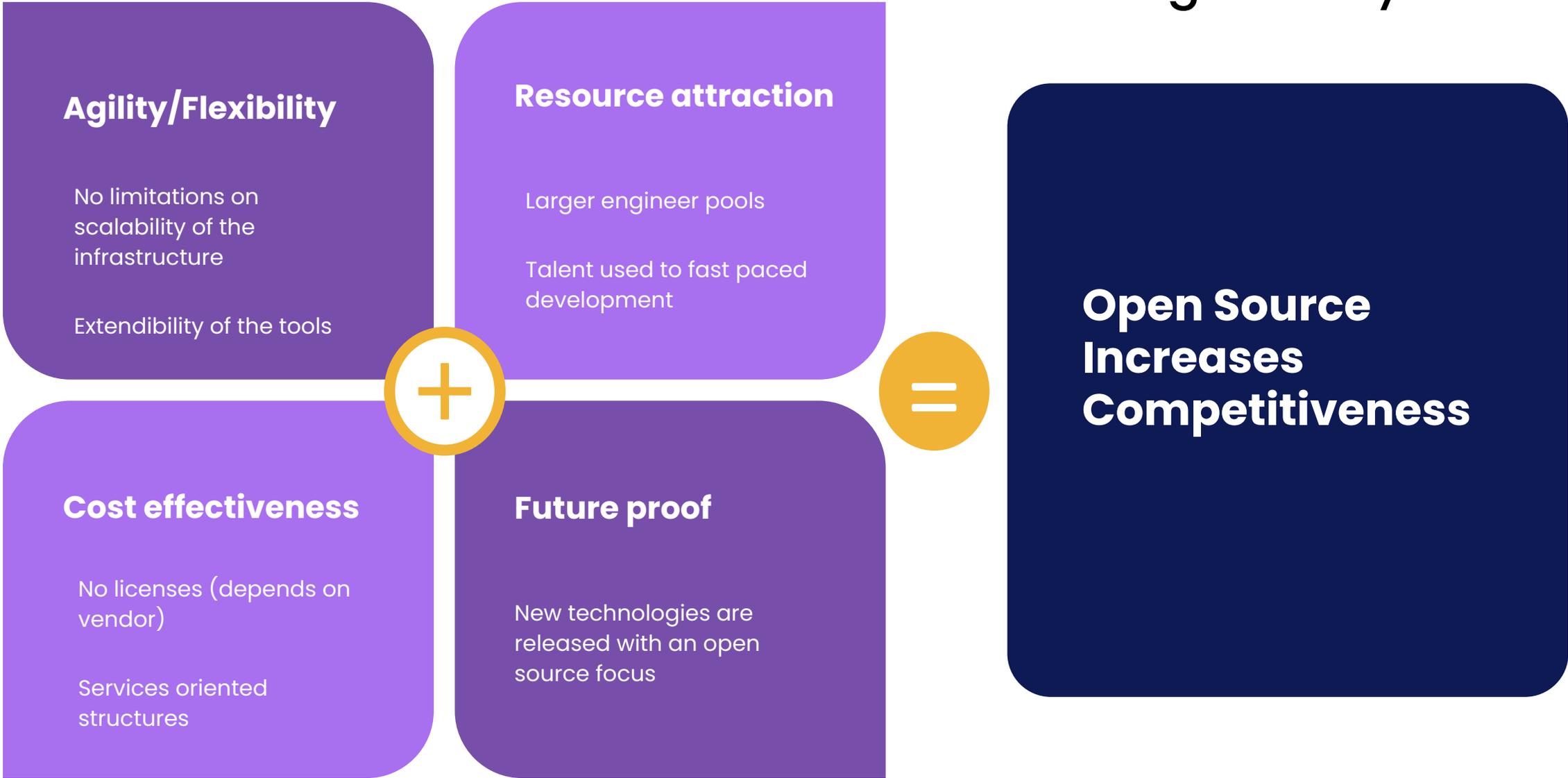 - Distributed: Cassandra, CouchDB, TiDB, CockroachDB

Risks

- Migrations are hard but manageable
- Perceived security
- A complete database technology migration is hardly ever possible
- Knowledge training is important
- Human capital constraints
- Watch out not to migrate from Closed source to another Lockin
- Timelines can be long

PERCONA

How new vendors/orgs can help you

- Training your staff of existing DBA's
- 24/7 support on your open source database stack
- Leveraging external team to help you augment your capabilities
- Migrate the dataset and functionalities
- Auditing infrastructure

PERCONA

So again Why?

Agility/Flexibility

No limitations on scalability of the infrastructure

Extendibility of the tools

Resource attraction

Larger engineer pools

Talent used to fast paced development

Cost effectiveness

No licenses (depends on vendor)

Services oriented structures

Future proof

New technologies are released with an open source focus

+

=

Open Source Increases Competitiveness

Conclusions

Thank You!