Making choices:

What kind of relationship are you seeking with your database?

February 13, 2014

J.R. Arredondo
Director, Data Services Product Marketing
@jrarredondo

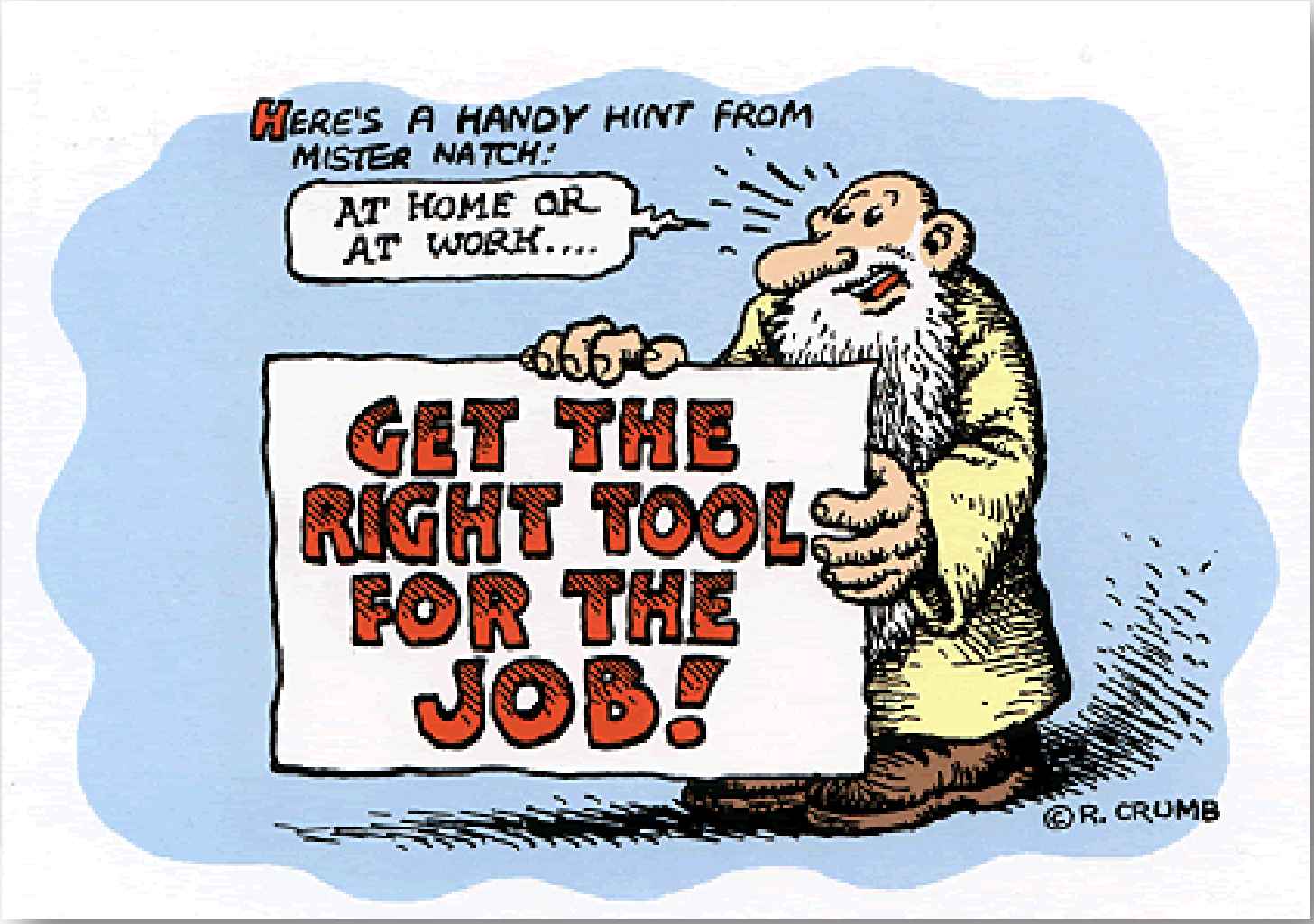
rackspace®
the open cloud company

This is your chance to leave the room!

You paid a lot of money to be here! Nobody will be offended if there is a better session for you!

Ballroom AB	• MLbase: Distributed Machine Learning Made Easy
Ballroom CD	• Real-time Analytics with Open Source Technologies
Ballroom E	• The Great Debate: Technology Creates More Jobs than it Destroys
GA Ballroom J	• Making Data Move: Stream Processing in Go
GA Ballroom K	• StatusWolf: Creating Dashboards That Don't Suck Using Art and Engineering
Mission City M	• Driving the Future of Smart Cities - How to Beat the Traffic
Ballroom F	• Real-Time Analytics with NewSQL: Why Hadoop is not enough
Ballroom H	• Think Big Data is the Answer? Think Again.

Common advice these days from smart people



This is what the they think you are about to do

(not that you are about to)

Let’s take a step back

rackspace®
the open cloud company

Databases are not simple, single purpose tools

What kind of relationship are you seeking with your database?

How did we get here?

rackspace®
the open cloud company

App development is changing

	Traditional apps (CRM, HR, Finance apps)	**Modern apps** (mobile, social, media, games)
Infrastructure	**Custom-built for the app**	**Programmable by the app**
Data	Mostly resides on premise	Mostly resides on cloud

Trend

Applications are becoming systems of engagement

	Traditional apps (CRM, HR, Finance apps)	**Modern apps** (mobile, social, media, games)
Characteristics of the system	**Systems of Record** **Highly structured** **Slow to change** **Transactional** **Stable** **Core to the business** **Not very social**	**Systems of Engagement** **Loosely structured** **Quick to adapt** **Conversational** **Dynamic and in flux** **Edge of the business** **Fundamentally social**
Data	Mostly resides on premise	Mostly resides on cloud

Trend

We are building different kinds of applications

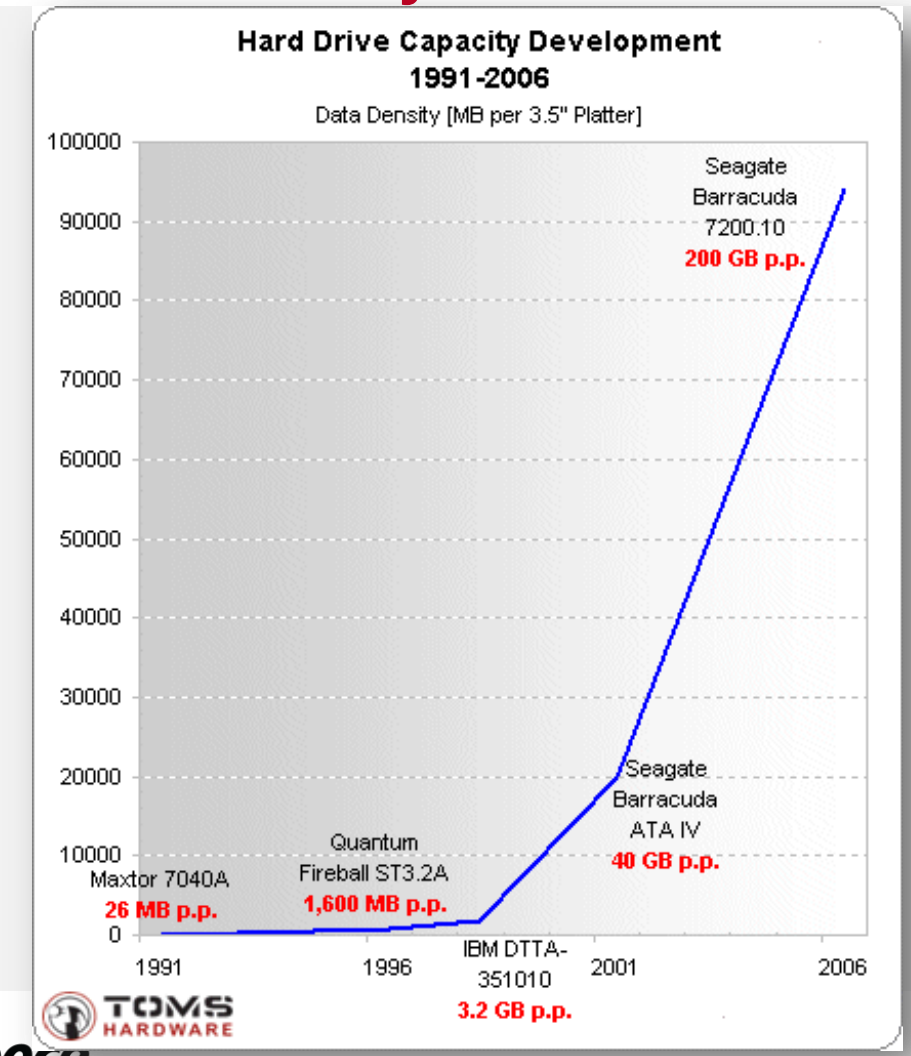
MEDIA | GAMING | M2M | MOBILE | SOCIAL

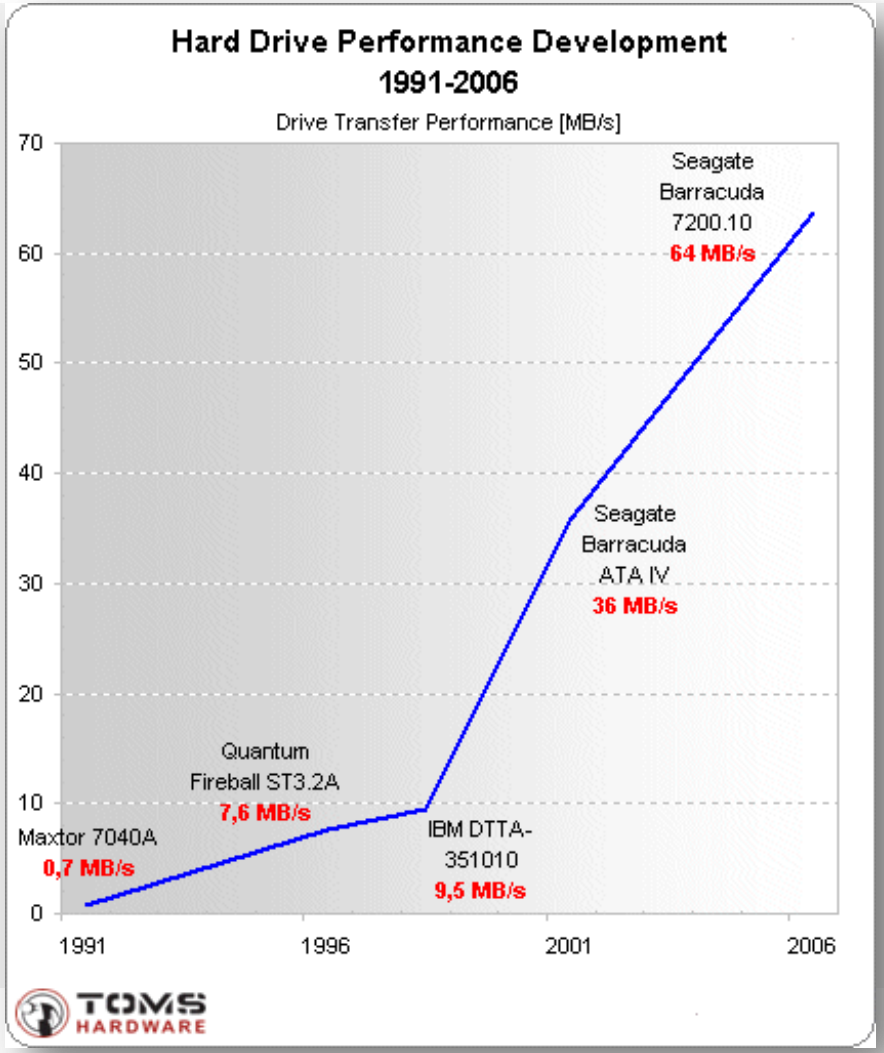
tumblr. Linked in facebook twitter Pinterest YouTube Broadcast Yourself

SOME UNIQUE SCENARIOS

Frequently written, rarely read
Binary files
Short term data
Multi-location access
Zero downtime needs
Dynamic or object oriented models
Trying to avoid RAID / storage limits
High speed data retrieval needs
Large files

In the 15 year period before 2006, storage density increased 10,000x, but performance only increased about 100x





rackspace
the open cloud company

Source: “15 Years of Hard Drive History: Capacities outran performance” (November 27, 2006)
http://www.tomshardware.com/reviews/15-years-of-hard-drive-history,1368-6.html

As a result, a revolution ensued in the world of Data Services

There are about 150+ choices **just** in the “NoSQL” subset

rackspace
the open cloud company

How should we pick?

rackspace®
the open cloud company

What are you, fundamentally, trying to do?

Transact

Improve an implementation

Analyze

Or both (operational intelligence!)

Understand the personality of your database

Let's use these examples (things may be different in your implementation)

Relational	Documents	Key-value	Distributed large sets
Data Integrity SQL	Flexible Schema Scale	Fast Retrieval Data structures	Distributed Processing Big Data
(SQL)	(MongoDB)	(Redis)	(Hadoop)

rackspace
the open cloud company

Transacting

rackspace®
the open cloud company

Relational databases

They literally saved the world from running major operations using paper

Strengths

- **Data integrity** through data types and semantic rules
  - AGE >= 0
  - Person must have a NAME
- **Querying**
- **Aggregation**
- **SQL**

```
SELECT SUM(VALUE)
FROM CAR
GROUP BY MODEL;
```

“Weaknesses”

- **Complex development** as developer needs to map relational model with object oriented code
- **Complexity grows** exponentially as relational model grows
- **Difficult to scale**
- **Expensive** (hardware, software)

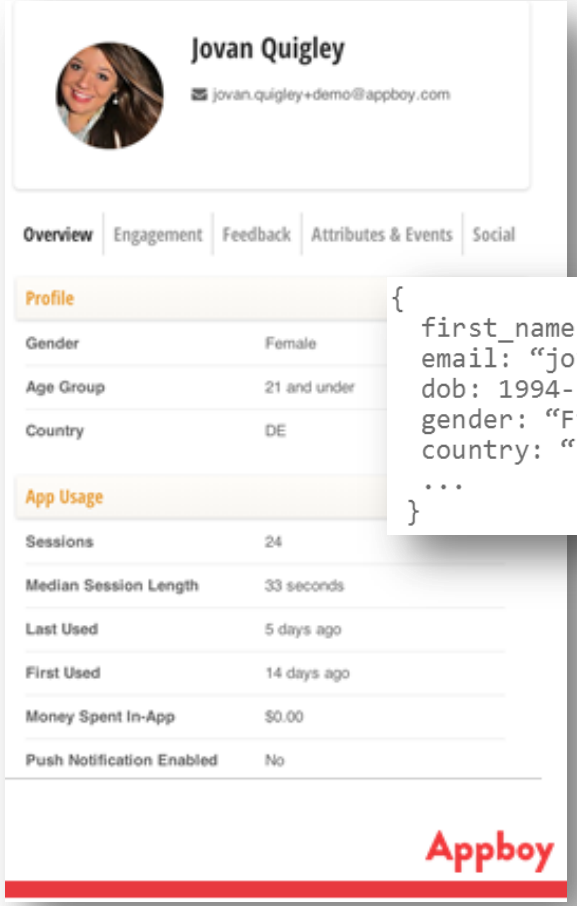
If your operation depends on the integrity of your business rules, the relational model rules.

Scaling is a little difficult.

rackspace
the open cloud company

Flexibility of data model (and its problems) with document databases

Appboy: App marketing automation platform for mobile apps

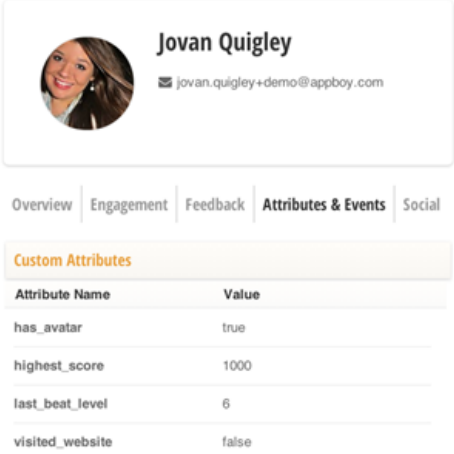



```
{
  first_name: "Jovan",
  email: "jovan+demo@appboy.com",
  dob: 1994-10-24,
  gender: "F",
  country: "DE",
  ...
}
```

Extensible User Profiles

Custom attributes can go alongside other fields!

```
{
  first_name: "Jovan",
  email: "jovan+demo@appboy.com",
  dob: 1994-10-24,
  gender: "F",
  custom: {
    has_avatar: true,
    highest_score: 1000,
    visited_website: false,
    ...
  },
  ...
}
```




```
db.users.find(…).update({$set: {
```

```
{ visited_website: true }
{ visited_website: "yes" }
```

Appboy

Courtesy of Jon Hyman, CIO and Co-Founder of Appboy

rackspace
the open cloud company

Tradeoffs

Data integrity
Business rules
Consistency
Transaction isolation
Atomicity

and

Rigidity

Flexibility of schema
Dynamic data models
Horizontal scale
Easier to get started

and

Inconsistency of data

It's good to understand the fundamental "theory"

What does your problem really need?

ACID

- **Atomicity**: A transactions either happens completely, or not at all
  - No partial transactions
- **Consistency**: Transactions end in a "valid" state
  - No violation of rules
- **Isolation**: Transaction appears as if it is the only thing happening to the database
  - Relaxed most times
  - Deals with phantom, dirty reads or non repeatable reads
- **Durability**: Committed transactions are permanent
  - Even after failure

BASE

- **Basically available**:
  - Supporting partial failures without complete system failure
  - Design as if users would end up in different partitions
- **Soft state**:
  - Things can be in flux for a little bit of time
- **Eventual consistency**:
  - Things right themselves

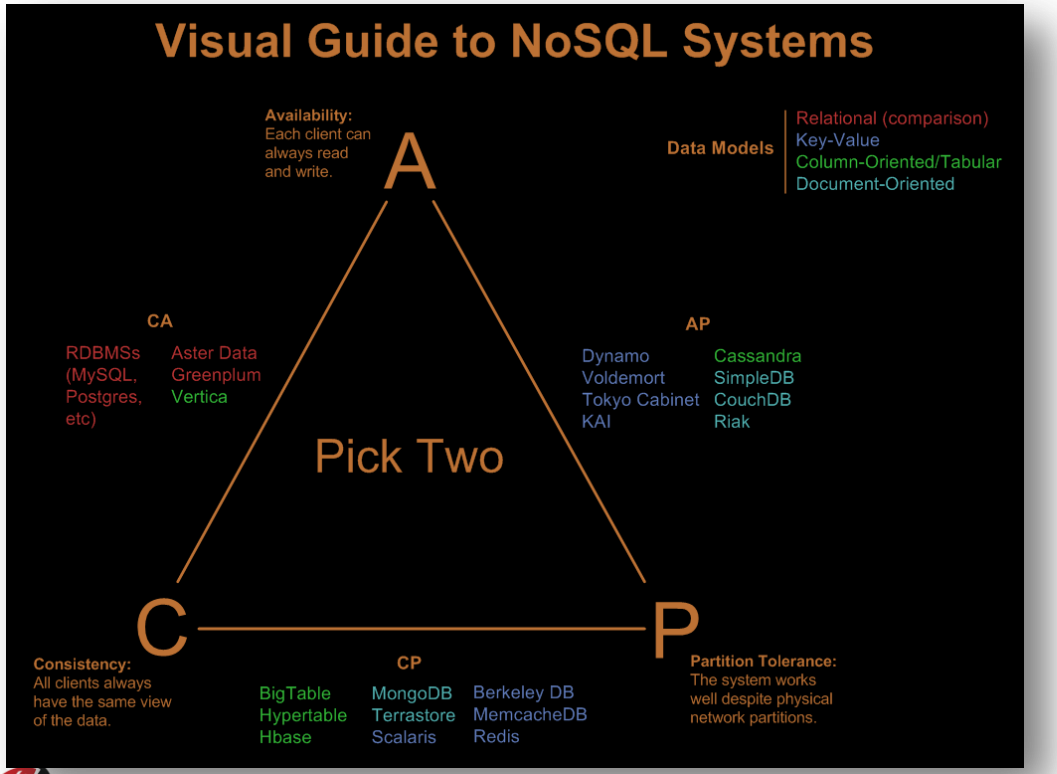
New ways of thinking:
Do customers really need to know the level of inventory of a product to place an order? Maybe all they want is to know that it is not zero

http://queue.acm.org/detail.cfm?id=1394128

Know your CAP, really

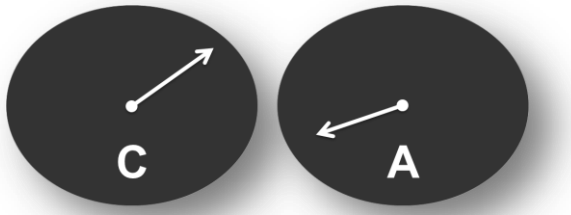
Consistency, Availability and Partition Tolerance

You can only have 2 out of 3 in CAP!



Wait! It's not that simple

- Partitions are not generally common
- Choosing Consistency or Availability is not final
- "It depends"
  - Maybe on user
  - Maybe on system
  - Maybe on type of data



- Just think:
  - How am I going to detect a problem in the network? (P)
  - How am I going to limit operations once I detect that?
  - How am I going to compensate to recover?

rackspace
the open cloud company

Hurst 2010 (http://blog.nahurst.com/visual-guide-to-nosql-systems)

Eric Brewer 2012 (http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed)

Sometimes…

… the answer is both

- Heavily used during weekends and at night
- Complex SQL queries
- "What are my friends drinking?"
- "Where can I find this beer?"

rackspace
the open cloud company

MySQL and MongoDB together

It's not one or the other

- What works best for the workflow?
  - MySQL worked best for reference data for us
  - Not everything moved to MongoDB

What stayed in MySQL?

Check-ins
Users
Relationships Data
Primary Datastore

What moved to MongoDB?

Activity Feed (Friend's Graph)
Recommendation Data
Location-based Check-ins

UNTAPPD
DRINK SOCIALLY

Courtesy of Greg Avola, CTO and Co-Founder, Untappd

What relationship are you seeking with your transactional database?

Be clear, upfront and intentional

Don’t let a database just happen to you

Analytics

This conference was full of good sessions on Hadoop analytics

Just two thoughts

rackspace®
the open cloud company

It is possible to be “too” data driven

We are dealing with people here

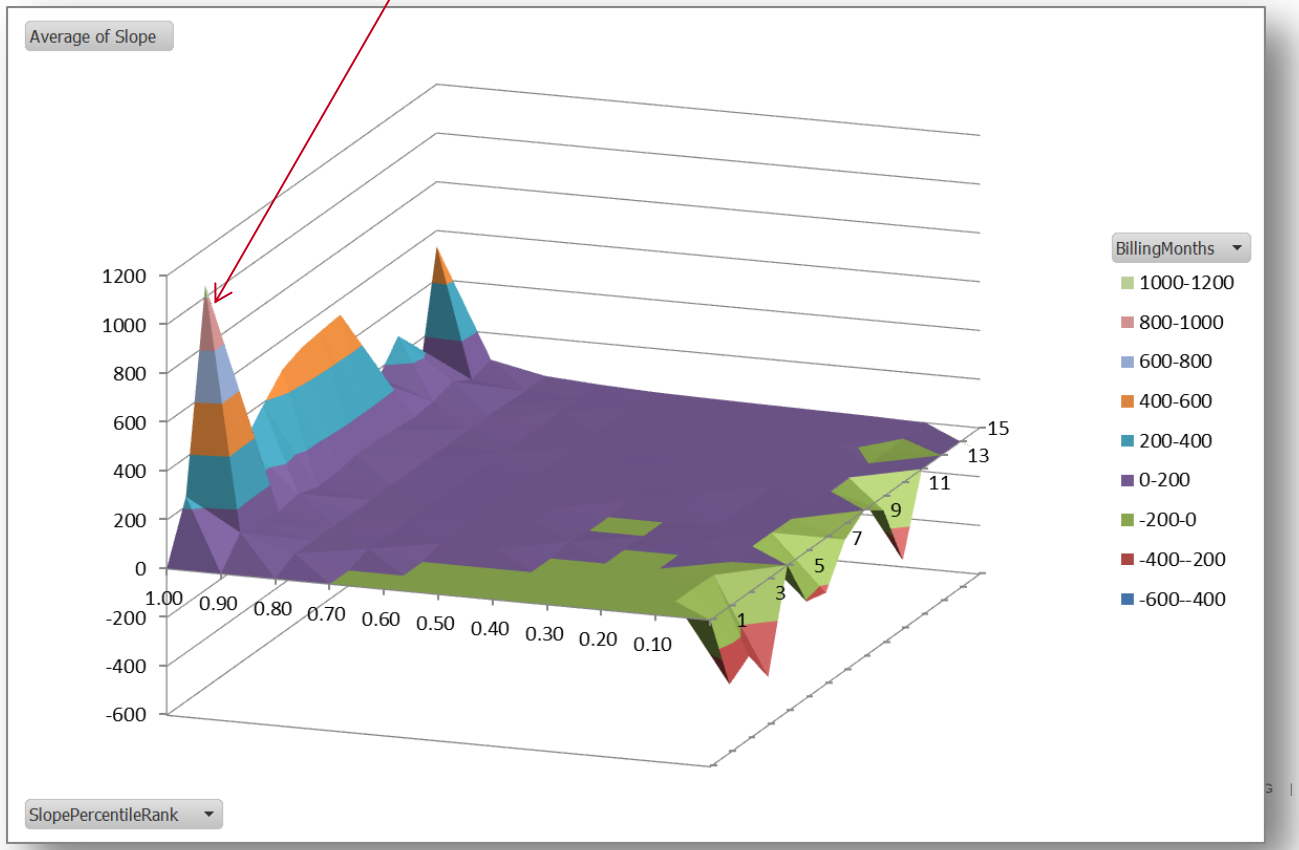
Data vs. Intuition

The role of visualizations and end user tools

Excel still rules in many places (*gasp*)

201203	Marker2	Trend	R
33440.59			
18007.59			
10901.37			
10721.84			
10077.15			
7597.64			
6355.67			
6025.06			
5400.66			
5291.82			
4785.9			
4521.8			

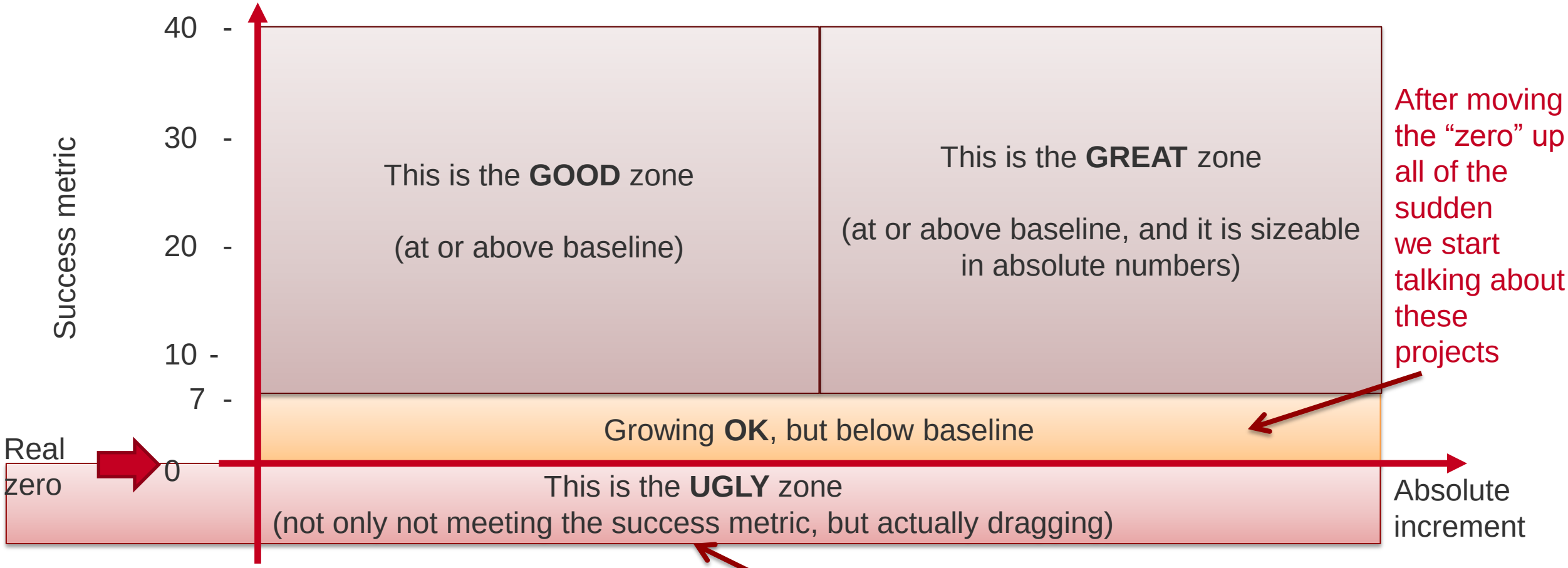
What is happening at this time?



rackspace
the open cloud company

Which things do you pay attention to?

"Redefining zero" changed the set of "things that we should discuss"



After moving the "zero" up all of the sudden we start talking about these projects

In meetings, everybody was first concerned with these projects

Data vs. Intuition: when do you need to use one vs. the other?

The iPhone would not exist if Apple would have had focus groups or made it a "data driven decision"

What relationship do you seek with your analytics backend, say, Hadoop + others?

Hadoop, please just be there for me, in the background

So that I can do what we really need to do in this place (drive insights to the business)

“Improving” an app

Just one simple example before we go

rackspace®
the open cloud company

The “ilities” and their cousins: Improving an app

These are some of the challenges indirectly related to data that we must deal with

- Stability
- Fit for core scenarios
- Configurability to different scenarios
- Integration with development languages
- Integration with other databases
- SQL compatibility
- End user vs. Developer skillset
- Conceptual changes
- Platform availability
- Data type and semantic needs
- Security

- Performance
- Scalability
- Consistency
- Resiliency
- Data model
- Flexibility
- Cost
- Training
- Tools availability
- Development experience

Really understand the personality of your database

Let's talk about Redis

"Redis is a cache"

- SET
- GET

Redis is a server for <u>data structures</u>

- Keys
- Strings
- Hashes
- Lists
- Sets / Sorted Sets
- Publish / Subscribe

Huge difference!

What relationship do you seek with your database?

One built on real understanding of what you bring to the table

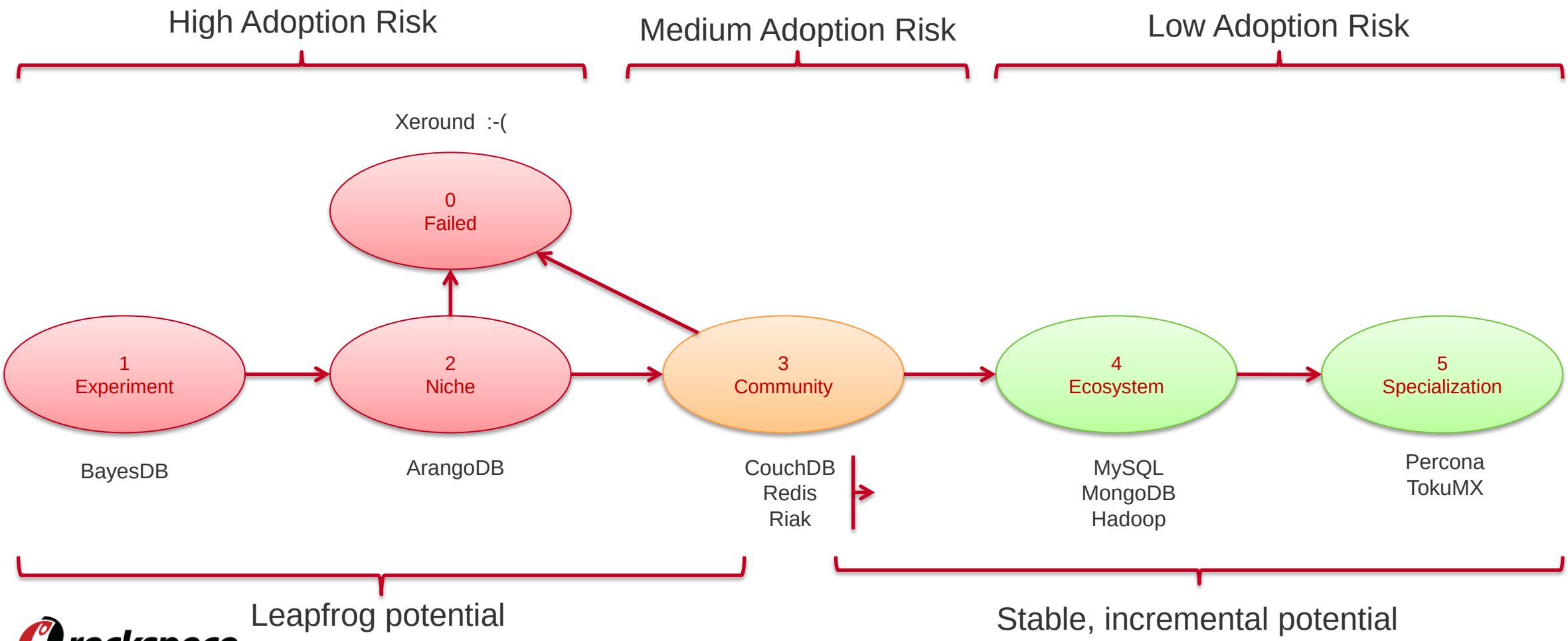
No first impressions please

Data Services also have a lifecycle

rackspace®
the open cloud company

Be aware of the "state machine" of Data Services evolution

Know what you are getting yourself into



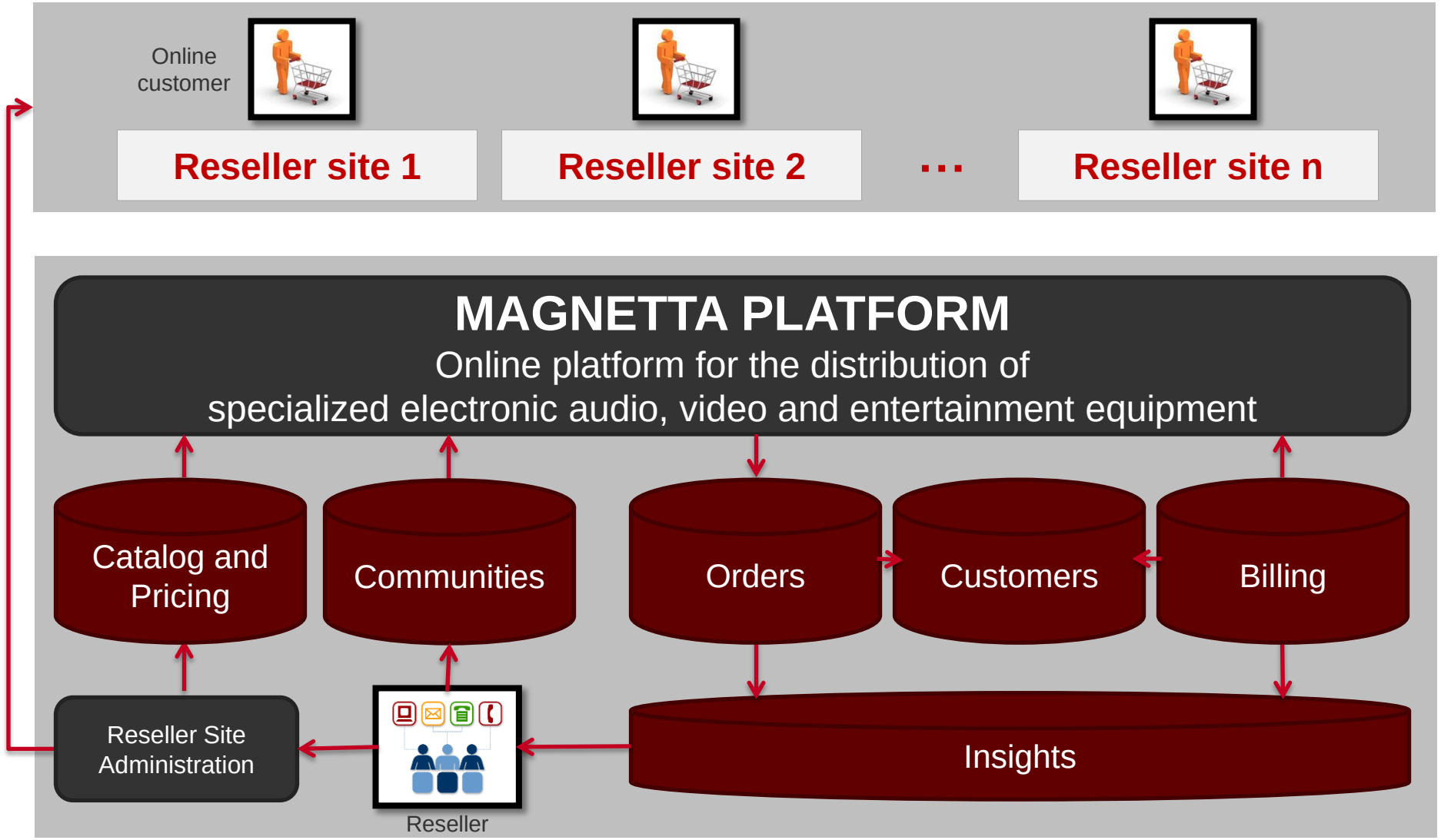
rackspace
the open cloud company

Simple things work

rackspace®
the open cloud company

Being pragmatic: seeing it all in practice

We will use a fictional eCommerce company called Magnetta



What kind of info could Magnetta store in each type of database?

(remember that **it always "depends"** and use it as the foundation for your data access layer)

Relational	Documents	Key-value	Distributed large sets
Customer contact **Reference data** Order Details (Ship To, Bill To SKU, Quantity, Price) Billing transactions Inventory Prices Member Info (user, pwd)	Customer relationships Notes / Social Partitions (shards) Promotional materials **Dynamic schemas** Statements **Product Catalog**, Images Product Configuration Personalized catalog Member Comments Product Reviews Product Q&As	Session info Cart Recent orders **Home page info** Latest comments Recommendations **Product "stars"** Upsell/Cross sell	**Customer attributes** (non PII, geo) **Sales history** Churn info Price history Social info Comments "NPS" **Recommendations** All kinds of analysis
(SQL)	(MongoDB)	(Redis)	(Hadoop)

Rackspace's vision is Data as a Service

From databases to data as a service

rackspace®
the open cloud company

From Database-as-a-Service to Data-as-a-Service

Focus on **building your app**, not managing databases

Highest value activity for your application

Build your application
(i.e. game, startup, mobile app, site)

PRESUMABLY YOU WANT TO BE FOCUSED HERE

This is the only job that YOU MUST DO without anybody's help because this is your intellectual property

Manage software infrastructure
(i.e. databases)

Manage hardware infrastructure

PRESUMABLY YOU DON'T WANT TO HAVE TO MANAGE DATABASES OR SERVERS

It only takes away from time building your application

The next vision for databases: Data-as-a-Service

Applications just access the **data as a service**, while the database is transparent

Highest value activity for your application

Build your application
and
manage your data

PRESUMABLY YOU WANT TO BE FOCUSED HERE

This is the only job that YOU MUST DO without anybody's help because this is your intellectual property

hostname, port number

Data

as a Service

The app just interacts with THE DATA

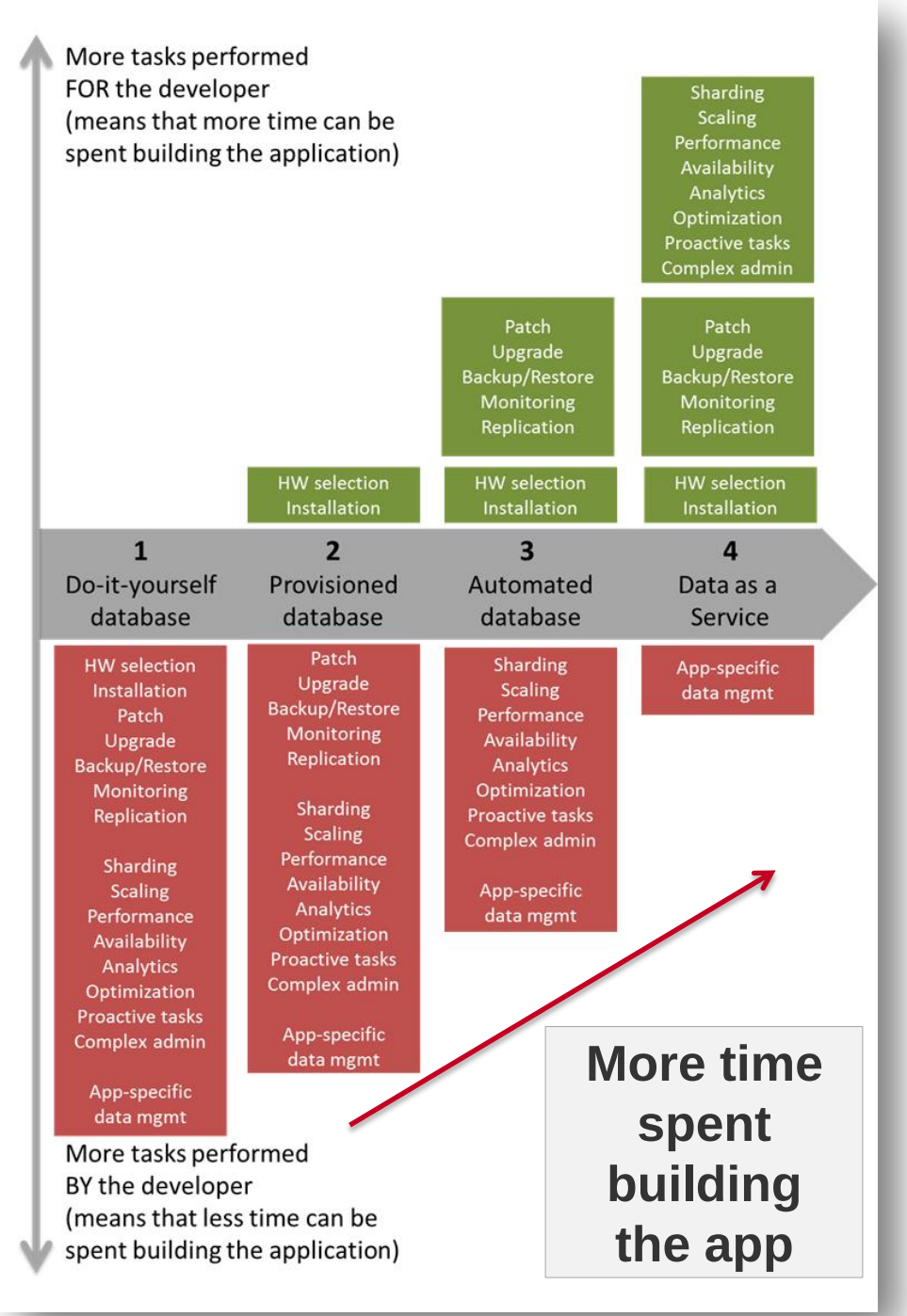
The application does not see the infrastructure

Towards transparent databases

Data-as-a-Service: more time building, less time managing databases

Four levels of DaaS transparency

- For some businesses, database or infrastructure management **IS core of the business**
- For most software-based businesses, database or infrastructure management represents time and resources **not spent building the application**
- **You must answer for yourself**: are you in the business of managing infrastructure, or in the business of [**your market here**]?



rackspace
the open cloud company

Source: "Choosing The Right Cloud Provider" (December 5, 2013)
http://www.rackspace.com/blog/choosing-the-right-cloud-provider-for-your-mongodb-database/

Data has mass and gravity: you need choices

(Or: “Divorces are expensive”)

Your Private Cloud on prem	Private Cloud at (your favorite provider)
Public Cloud	Managed Cloud

rackspace®
the open cloud company

Data Services at Rackspace

2 offerings in partnership with Hortonworks for Hadoop-based applications

2 acquisitions for MongoDB and Redis apps

Strong portfolio of traditional offerings

In Summary, databases are like relationships

Don’t let a database just happen to you (be intentional)

Don’t lose sight of the business goal

- Databases are best when invisible

Really understand the true nature of the database

Database management = time not spend building your business app

rackspace®
the open cloud company

THANK YOU

@jrarredondo

RACKSPACE® HOSTING | 5000 WALZEM ROAD | SAN ANTONIO, TX 78218
US SALES: 1-800-961-2888 | **US SUPPORT:** 1-800-961-4454 | WWW.**RACKSPACE**.COM