Noms

A new database for the Big Data World

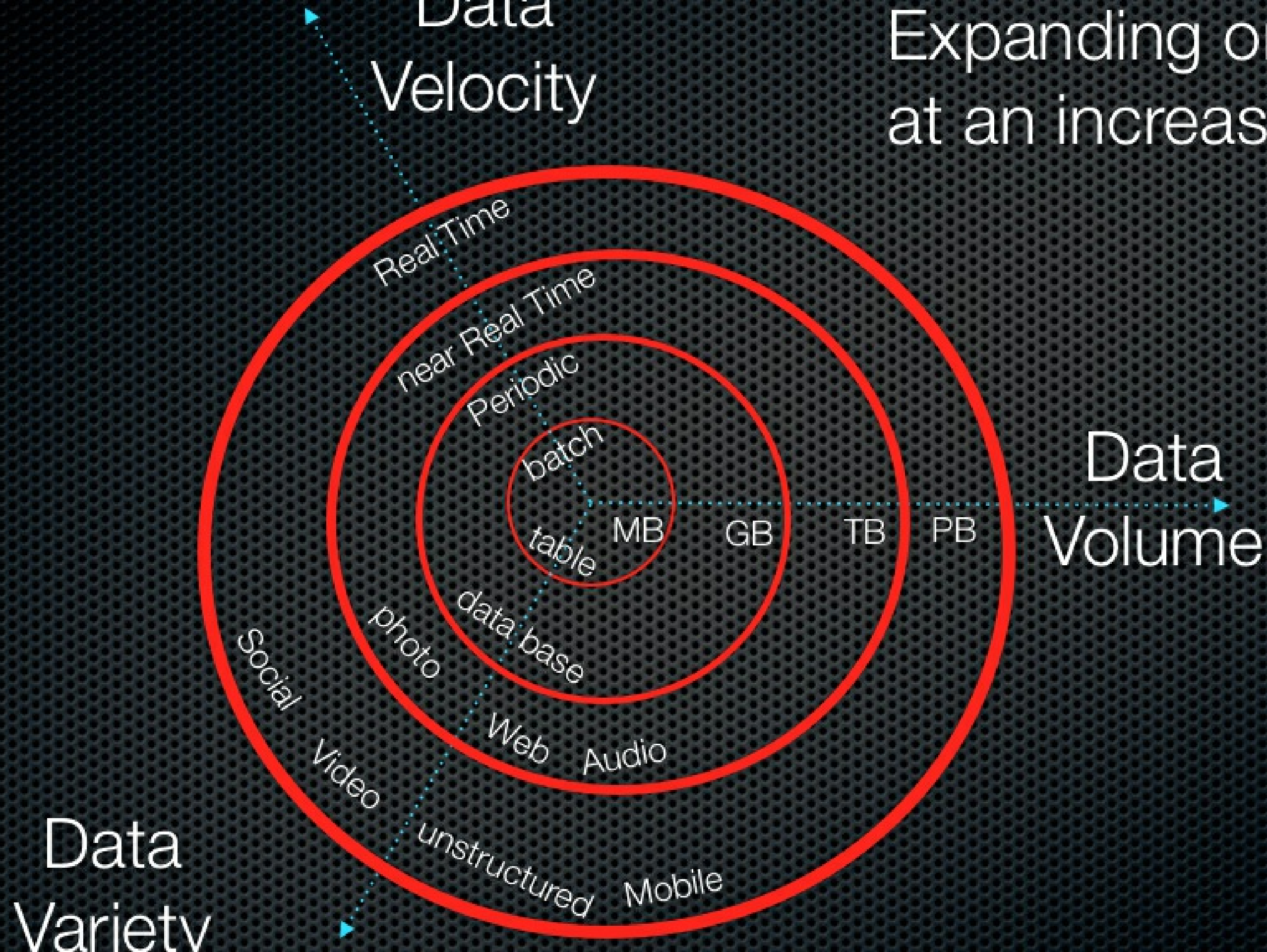### The Developers Conference

Porto Alegre 2016

Big Data

"Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them."

source: https://en.wikipedia.org/wiki/Big_data

3D Data Management: Controlling Data Volume, Velocity, and Variety

Doug Laney, 6th February 2001

today's challenges include...

sharing & storage

Introducing Noms

"Noms is a decentralized database based on ideas from Git."

Attic Labs

- Founders (Ex-Googlers): Rafael Weinstein / Aaron Boodman
- Raised $8.1M (mainly from Greylock Partners)
- Source code available @GitHub, developed in Go

About Noms

Decentralized

Data is modeled as a directed acyclic graph

Users can have their own copy of the database

Any peer is free to move forward independently

About Noms

Content-addressed

Intended to store data that does not change over time

Concept of content identifier

Duplicated data is automatically deduplicated

About Noms

Append-only

Data is never removed by default

About Noms

Strongly Typed

Every value has a type

Types are evaluated when you insert new data

Noms X Traditional Database

Noms	Traditional
Any peer's state is valid	Single source of truth
Historical records	Single point in time

Noms Concepts

- Databases / Datasets
- Chunks
- Types
  - Boolean, Number, List, Blob, etc
  - Allow data to be self-describing
  - Every chunk has a header that describes its type
  - Types can be verified by clients

Type Accretion

Widen (or narrow) the type of any container

```
Set<struct {name: String}>
```

becomes

```
Set<struct {name: String}> | struct {name: String, age: Number}>>
```

Project Status

Data Format

Core data format defined

Project Status

Completeness

We're just getting started

Project Status

API

Public API will continue to evolve

Noms CLI & SDK

An introduction to Noms CLI and JS SDK

Installing

Command tools can be downloaded from GitHub or build from source code (requires Go 1.6+)

https://github.com/attic-labs/noms

Serving a database

Databases uses Google LevelDB as backend and can be persistent or not (in-memory).

```
noms serve ldb:<database>
```

```
noms serve mem
```

Querying datasets

List of available datasets in a database

```
noms ds ldb:<database>
```

```
noms ds http://<server:port>
```

Syncing datasets within local and remote databases

Similar to git push / pull

```
noms sync <origin> <destination>
```

Pulling from remote to local

```
noms sync http://www.acme.com:8000::ds ldb:db::ds
```

Pushing from local to remote

```
noms sync ldb:db::ds http://www.acme.com:8000::ds
```

Historical commit log of a dataset

```
noms log http://<server:port>::<dataset>
```

```
commit nb28vpmt3u0of37u1hs2lis5nqo7942j
Parent: 4niqn6lu9kuq6i35ovq26vqfaov60vs3
(root) {
+   {
+     name: "Gustavo",
+     phone: "9876-5432",
+   }
}

commit 4niqn6lu9kuq6i35ovq26vqfaov60vs3
Parent: None
```

Displaying meta data and data values

```
noms show http://<server:port>::[<dataset>][#<commit>]
```

```
struct Commit {
   meta: struct {},
   parents: Set<Ref<Cycle<0>>>,
   value: List<struct {
      given: String,
      male: Bool,
   }>,
}({
   meta:  {},
   parents: {},
   value: [  // 4 items
      {
         given: "Rickon",
         male: true,
      },
      {
         given: "Bran",
         male: true,
      },
      {
         given: "Arya",
         male: false,
```

Diffing two commits

```
noms diff http://<server:port>::#<commit-1> http://<server:port>::#<commit-2>
```

```
.parents {
-   nb28vpmt3u0of37u1hs2lis5nqo7942j
+   4niqn6lu9kuq6i35ovq26vqfaov60vs3
}
.value {
-   {
-     name: "Luiz",
-     phone: "1234-5678",
-   }
}
```

JavaScript SDK

Allows a developer to work with databases, datasets, commits, data, etc.

npm install @attic/noms

https://github.com/attic-labs/noms/blob/master/doc/js-tour.md

DEMO

References

- Slides / Sample code
- Noms @ GitHub
- Introduction to Noms
- JavaScript API
- NPM API Package
- Slack Team
- Attic.io

Thanks!!

Gustavo Coelho
Software Engineer @ Hewlett Packard Enterprise

Email: gcoelho@gcoelho.com
Twitter: @gafcoelho
GitHub: github.com/gcoelho

Luiz Svoboda
Software Engineer @ Hewlett Packard Enterprise

Email: luizek@gmail.com
Twitter: @luizsvoboda
GitHub: github.com/lhsvobodaj