

# Inside Infogami

Anand Chitipothu

# Inside Infogami

- Infobase - Storage
- Infogami - presentation

Infobase

# What is Infobase?

- Storage

# Why not SQL?

- Schema changes are painful
- No revision history
- Type unification is tricky
- caching and scalability issues

# Infobase

- Stores objects
- Object is a list of key-value pairs
- Every object has an associated type

Key	“b/tomsawyer”
type	“type/book”
title	“The Adventures of Tom Sawyer”
authors	[“a/mark_twain”]

Key	“a/mark_twain”
type	“type/author”
name	“Mark Twain”

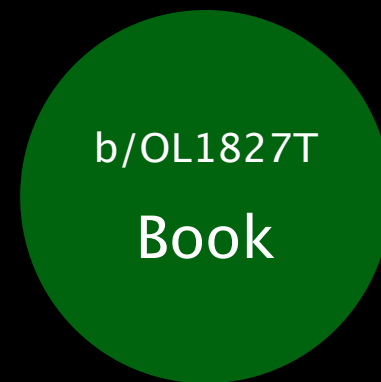
# Infobase

- type is also object
- type specifies the expected properties

key	"type/book"
type	"type/type"
properties	["type/book/title", "type/book/authors"]

key	"type/book/authors"
type	"type/property"
name	"authors"
expected_type	"type/author"
unique	FALSE

# Version Control



title	"The Adventures of Tom Sawyer"
author	a/Mark_Twain
subject	s/Fiction
subject	s/Americana
published	1876
language	"en"

**aaronsw**  
2007-11-05  
initial create

title	"The Adventures of Tom Sawyer"
author	a/Mark_Twain
subject	s/Fiction
subject	s/Americana
published	1876
language	"en"
pages	275

**brewster**  
2007-12-04  
add pages

title	"The Adventures of Tom Sawyer"
author	a/Mark_Twain
subject	s/Fiction
subject	s/Americana
published	1876
language	"en"
pages	275
folk_novel_id	F92982

**folkfan47**  
2008-01-02  
add folk\_novel\_id



- Every thing is object
- Schema updates are trivial
- All changes are tracked
- Time travel

# Infobase API

- JSON API to read and write objects
- Read
  - `/get/path`
  - `/things`
  - `/versions`
- Write
  - `/write`

# User Management

- Infobase has built-in support for user accounts and user groups
  - `/account/register`
  - `/account/login`

# Permission System

- Every object can have a permission associated with it to determine who can read and who can modify
- If permission is not specified, permission is inherited from the parent.

# Caching

- Store objects and queries in the cache
- On save
  - invalidate the saved object
  - invalidate the matching queries

# Replication?

- Find the recently modified versions in the master
- fetch them and update the slave



# Infobase

- Storage
- Read Write API
- User Management
- Permission System
- Caching
- Replication?

Infogami



# Infogami

- Structured wiki
- Uses Infobase as storage
- Plugin architecture
- Templates are stored in wiki

# Uses Infobase

- URL mapping for free
- User accounts for free
- Permission system for free
- Caching for free

# Templates

- Templates for layout
- Templates for view and edit every type of page

# Templates in the wiki

- Web designer can work away
- Skins come for free
- Users can create skins
- Tied to data, not code (time travel!)

# Plugin Architecture

- Extensions can be added as plugins to infogami
- Lot of hooks

# Scaling

- Run multiple web servers working with the same infobase server
- Replicate infobase if required

Questions?

Thanks!