

Diaspora

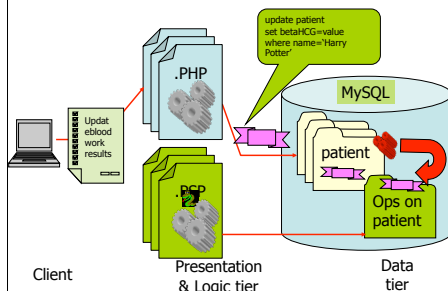
Goal

- Extend critical information across company boundaries
 - Orthogonal & builds on identity management (enabling portability of identity information across autonomous security domains)
- Enable users of one domain to access data or systems from other domains seamlessly / with no explicit user administration effort

Insight

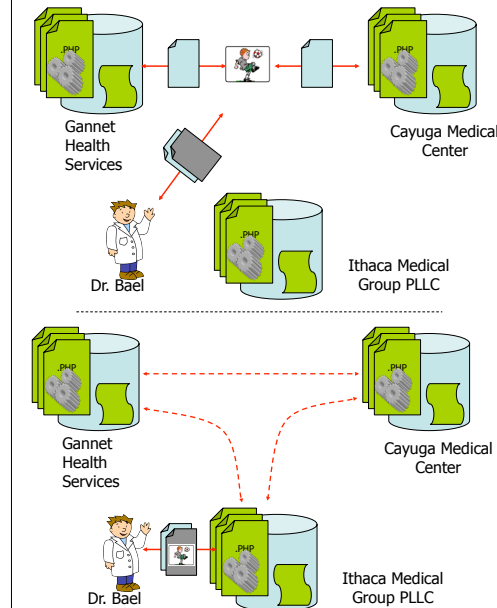
Diaspora as a webapp itself

- A collection of choreographed web apps on each site
 - Store data structures pertaining to own web application logic
 - Access local domain data and make it available to peer web apps on different sites / different domains



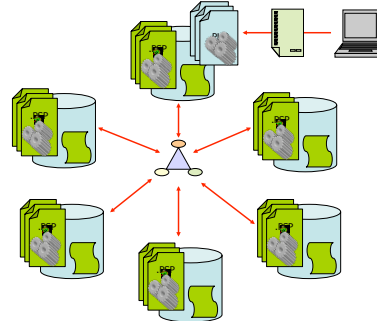
- Diaspora internals
 - Triggers are installed on target databases
 - Every modification is logged into meta tables
 - Diaspora peer webapps "talk about" the meta tables
 - Issues
 - Database privacy
 - Consistency (Harry Potter != H. Potter)

The problem



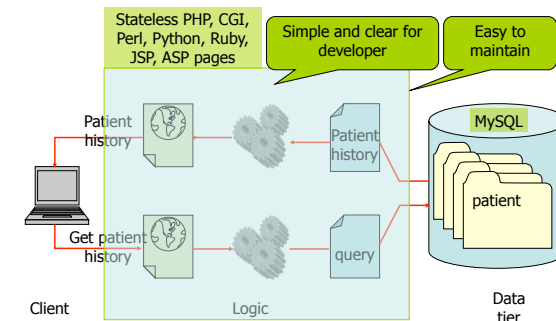
Coordinator entity – Administrator

- A distributed system in itself handling: replication, leader election, failure detection, rollback & recovery
- Stores its data replicated in the meta tables



Approach

- Seamlessly interact with existing systems / minimize adoption efforts
- Assumptions
 - Data stored in relational repositories
 - Data can be made readily available by means of standard web interfaces
 - In particular stateless "active" pages



Usage Example: Automatic Replication of Database Tables

- Web hosting companies offer
 - Restricted user account (MySQL + FTP)
 - MySQL, PHP, Perl, Python, Ruby as lightweight Apache modules
- Owner of multiple accounts / over heterogeneous hosting companies
 - Leverage all for reliability, replicate
 - May only use the above mentioned model

