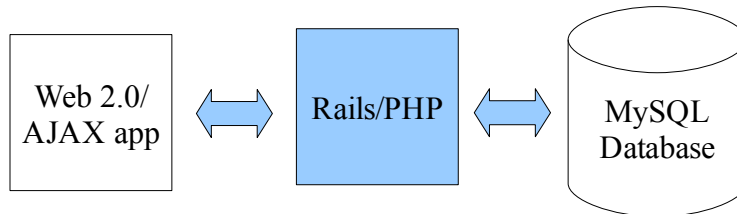


NextDB.net – Features and Benefits

Web 2.0 and AJAX

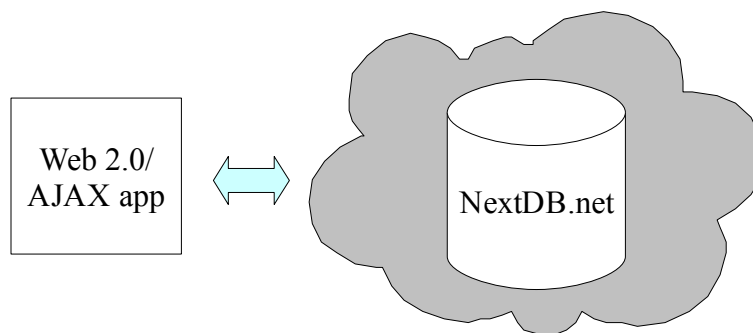
Today's web applications rely heavily on JavaScript. Web apps that use JavaScript to talk to databases are called AJAX apps. Examples of AJAX apps include *Gmail*, *Facebook*, *Youtube*. These rich-client AJAX apps are faster, and work better than oldschool web apps, but there is a hidden cost: the translation layer between the browser and the database. AJAX applications still get their data from old fashioned databases, so while the user experiences benefits from AJAX (on the front end), the back end costs of development and hosting are high.



Rails/PHP is a painful part of an AJAX application

NextDB.net Removes the Hidden Programming Costs

NextDB.net is a hosted relational database that removes the hidden costs of serverside programming. NextDB.net has a pure JavaScript programming interface so the AJAX programmer doesn't need a PHP or Rails server. This is unique to NextDB amongst all databases on the market. The developer simply logs into NextDB.net, creates a database, and starts writing JavaScript using his favorite JavaScript tools and libraries. There's no new programming language or environment to present a barrier to entry. NextDB eliminates hidden cost of an AJAX application--the pain of custom-written PHP, Perl, or Ruby.



NextDB.net is hosted in the cloud

Scalability and Performance

Traditional databases require a knowledgeable Database Administrator to keep the system optimized. A database like MySQL database will bog down as the size of the database grows unless its indexes are properly created and tuned to adapt to the changing needs of the application. NextDB replaces this labor intensive methodology with an automated indexing system, creating an advanced indexing plan uniquely tuned to the specific needs of an individual application. This automated indexing system is unique amongst relational databases, and is made possible by the fact that NextDB stores and compiles all the application's queries.

Convenience and Ease of Use

It's hard to conceive of a database that is easier to use than NextDB.net. The ease of use stems from the fact that NextDB.net requires no software installation, no server setup, and no proprietary development environment. No other relational database on the market can claim a web-based user interface, zero installation, and secure AJAX API. NextDB also includes, as native features, many capabilities that have traditionally required the integration of third party software. Just to name a few: CAPTCHA, email, and usage charts. NextDB even allows developers to easily share their data models with other programmers. NextDB's advanced query language, NextQuery will look familiar to SQL programmers, but is designed to be easy to master.

Security

Security issues plague databases. A common issue is the "SQL injection attack" in which a hacker is able to gain access total access to the database, and even the entire server. NextDB was designed around a patent-pending security paradigm designed specifically for a hosted database. Through tight integration with the secure NextQuery database language, NextDB prevents injection attacks and creates a digitally watermarked and encrypted "ticket" for data that travels to the browser. NextDB uses the Advanced Encryption Standard (AES) which is stronger than DES, and which in June 2003, the US Government announced may be used to protect classified information. NextDB's patent-pending "round trip" security model uses these encrypted tickets, called Secure Result Identifiers (SURID) to insure the integrity of data in the hostile JavaScript environment. This level of "web safety" is unique to NextDB amongst all relational databases.

Compare to Amazon Web Services

Amazon's SimpleDB cannot be accessed directly from JavaScript. This is because Amazon Web Services (AWS) requires every request to be signed by the developer's secret key. Because JavaScript is not a secure environment, the secret key cannot be used in the browser. This means you need to setup a secure server to sign all your requests which creates a bottleneck, and requires serverside programming. Amazon is a "server to server" environment, not a client-server environment. AWS's SimpleDB lacks the basic relational capabilities of a MySQL, Oracle, or NextDB. Any experienced developer knows that these features exist for performance and data integrity and attempts to compensate for a lack of these basic features leads to dangerous hacks and data inconsistencies.