

What is a spatial database?

Why would you want one?

PostGIS extends PostgreSQL to make it a spatial database.

Why would I want PostgreSQL

Scalable in two ways: technically and economically.

Economic scalability gives PostgreSQL / PostGIS great price/performance.

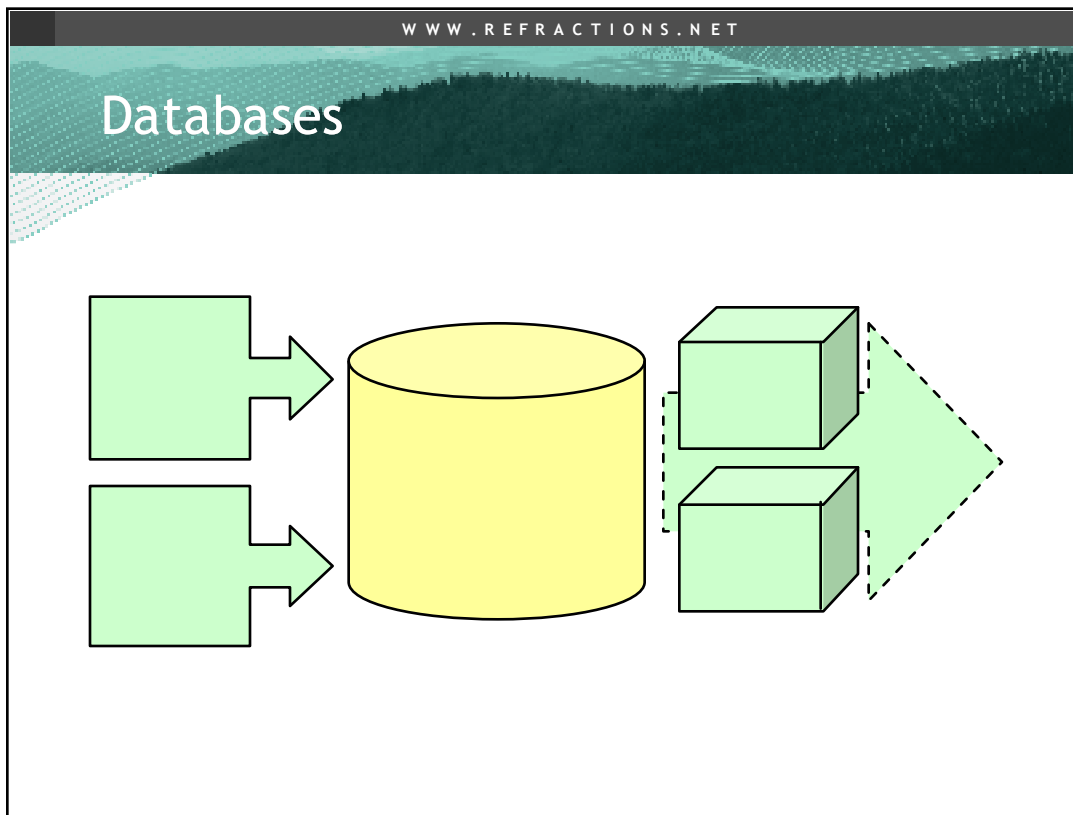
So if you transition to PostGIS, you can either lower your costs for the same performance, improve your performance for the same price, or even do both!

WWW.REFRACTIONS.NET

The Beginning

- PostGIS is an open source geospatial extension for PostgreSQL
- PostgreSQL is an open source relational database management system (RDBMS)
- A relational database management system is... ?

Let's start from first principles...



The relational database is the big yellow cylinder that sits in the middle of all our systems architecture diagrams. The reason it is always there, usually in the middle, is because it performs a critical role in systems architecture, providing a safe, random access way of storing large amounts of data.

Geographic information systems, manipulate INFORMATION, and that information has to reside somewhere.

It should be EASY TO ACCESS the data, with a STANDARD QUERY SYSTEM, and MULTIPLE USERS should be able to read and write that data over the network.

It should be FAST TO ACCESS search the data for particular pieces.

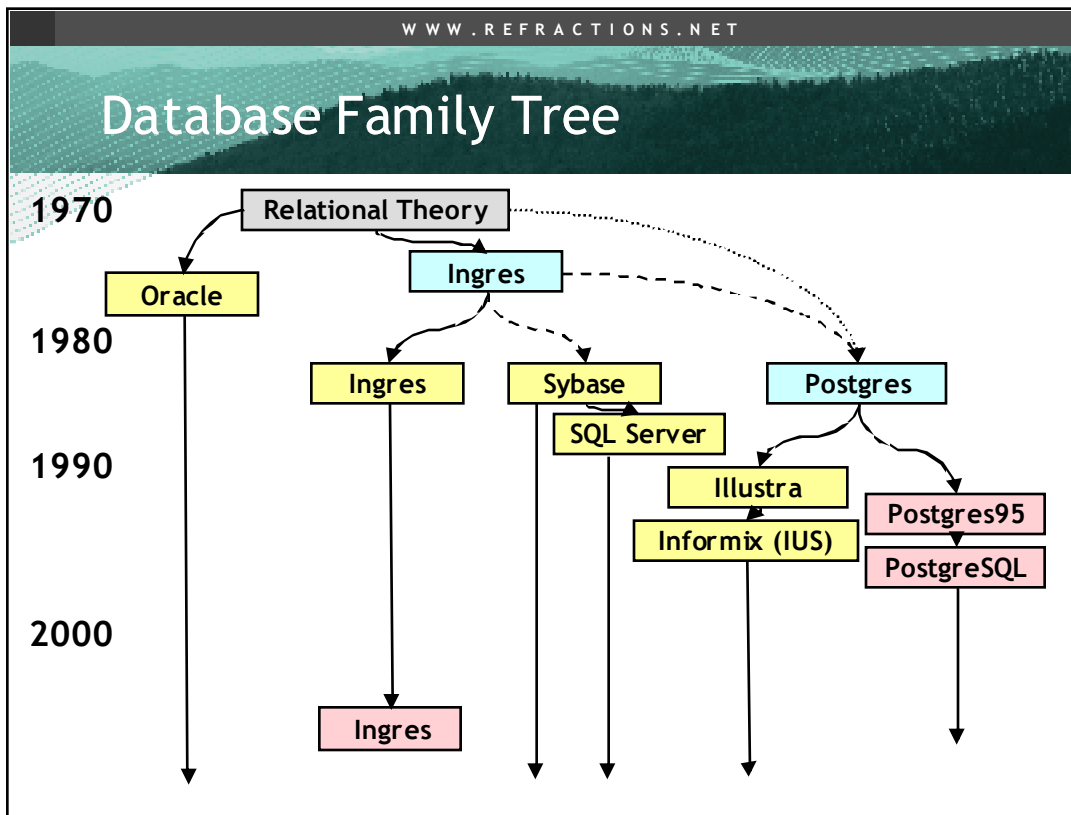
It should be SAFE from systems failures.

Michael Stonebraker



**Database researcher at University of California
Used teams of graduate students to create real database implementations**

- Ingres (1974-1984)
- Postgres (1985-1993)
- Cohera (1997-2001)
- Aurora (Ongoing)
- C-Store (Ongoing)



The **relational model** for [database](#) management is a [database model](#) based on [predicate logic](#) and [set theory](#). It was first formulated and proposed in 1969 by [Edgar Codd](#) with aims that included avoiding, without loss of completeness, the need to write computer programs to express database queries and enforce database integrity constraints. "Relation" is a mathematical term for "table", and thus "relational" roughly means "based on tables". It does not refer to the links or "keys" between tables, contrary to popular belief. Codd's original description of the relational model, "*Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*", was published in an IBM Research Report in 1969. A revised version of this paper, the highly acclaimed "*A Relational Model of Data for Large Shared Data Banks*", was published in [Communications of the ACM](#) the following year.

physical storage methods. This system has been standard ever since.

Oracle 1977, Software Development Labs, 1979 renamed Relational Software, 1983 renamed Oracle Corporation

1974 Ingres = Interactive Graphics REtrieval System

Originally a GIS proposal! Stonebraker and Eugene Wong.

Stonebreaker et al decided to pursue a relational database project instead.

Used GIS money as seed to land further DoD money.

1980, Relational Technology, 1985 renamed Ingres Corp

1990, Purchased by ASK

1994, Purchased by Computer Associates

2004, Open sourced by Computer Associates

2005, Spun back out as Ingres Corporation

1984 Sybase Founded

1992 SQL Server

1986-1993 Postgres

WWW.REFRACTIONS.NET

What is an RDBMS?

- System that provides for multi-user random access of potentially very large quantities of data.
- Modeling language (DDL)
- Query language (SQL)
- Transaction guarantees (ACID)
 - Atomicity, Consistency, Isolation, Durability

In general, a RDBMS is a software system that provides a short list of core functionalities. These are features that exist in every RDBMS worthy of the name. In particular, a database without ACID data integrity guarantees is less an RDBMS than a file management system that happens to use SQL (e.g. MySQL using MyISAM tables).

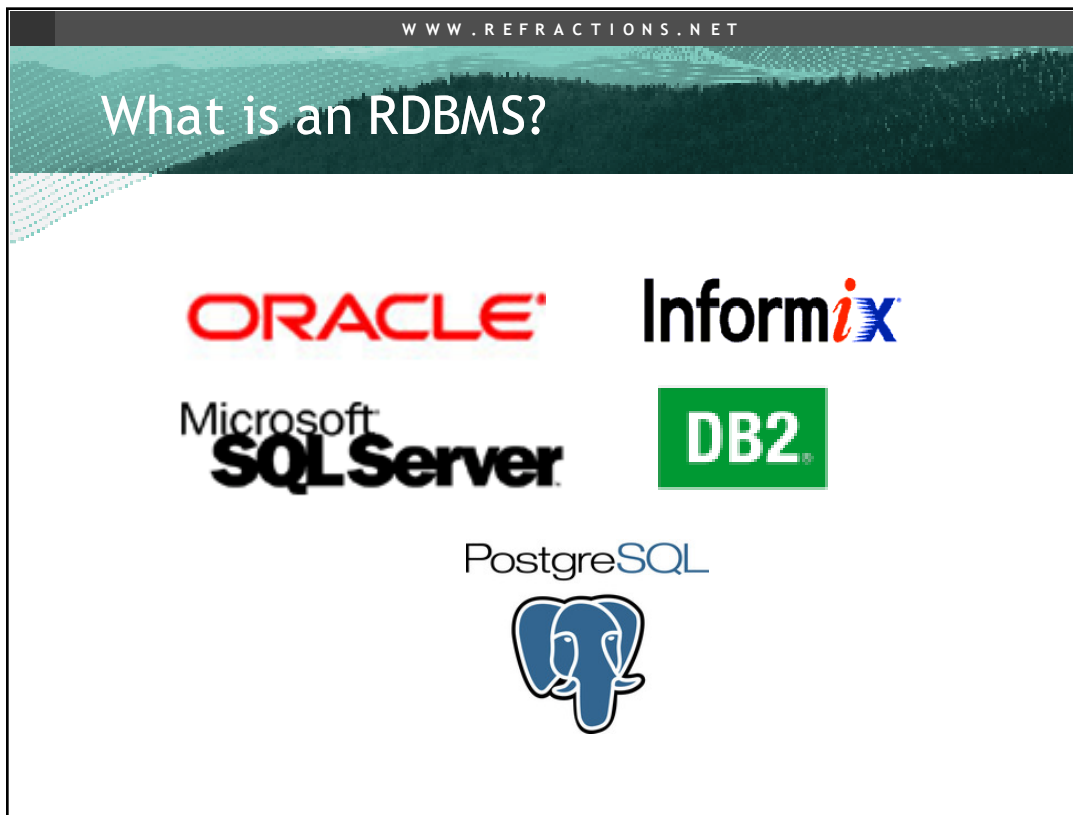
The definition of ACID is:

Atomicity refers to the ability of the DBMS to guarantee that either all of the tasks of a transaction are performed or none of them are. The transfer of funds can be completed or it can fail for a multitude of reasons, but atomicity guarantees that one account won't be debited if the other is not credited as well.

- Consistency refers to the database being in a legal state when the transaction begins and when it ends. This means that a transaction can't break the rules, or integrity constraints, of the database. If an integrity constraint states that all accounts must have a positive balance, then any transaction violating this rule will be aborted.

- Isolation refers to the ability of the application to make operations in a transaction appear isolated from all other operations. This means that no operation outside the transaction can ever see the data in an intermediate state; a bank manager can see the transferred funds on one account or the other, but never on both—even if she ran her query while the transfer was still being processed. More formally, isolation means the transaction history (or schedule) is serializable. For performance reasons, this ability is the most often relaxed constraint. See the isolation article for more details.

- Durability refers to the guarantee that once the user has been notified of success, the transaction will persist, and not be undone. This means it will survive system failure, and that the database system has checked the integrity constraints and won't need to abort the transaction. Typically, all transactions are written into a log that can be played back to recreate the system to its state right before the failure. A transaction can only be deemed committed after it is safely in the log.



Here are some RDBMS products you may have heard of. Note that ArcSDE is not on this list, because it is not a database, spatial or otherwise. It is, as even its marketing material notes, “spatial middleware”, that sits on top of a real database. The database provides the transactional guarantees and storage, and ArcSDE translates “spatial” to clients that know how to speak to ArcSDE (other ESRI products, anything implementing the SDE client libraries, and so on).

Of these database, only SQLServer does not have a native spatial data type.

Recently Microsoft announced an upcoming spatial capability for SQLServer, so in a couple years all these databases will be able to claim true native spatial data support.

WWW.REFRACTIONS.NET

What is PostgreSQL?

- Second generation RDBMS
- Michael Stonebraker
UC Berkeley
 - Database researcher
 - Ingres, 1977-1985
 - Prove relational theory
 - Postgres, 1986-1994
 - Extend and improve

In “The Design of Postgres” (1986), Stonebraker gave his goals for his next research database:

- Provide better support for complex objects
 - Move beyond strings/numbers/dates, to geometry, images, multi-media. Postgres included a run-time type/function catalogue that allows new types to be added without recompiling or changing the core software.
- Provide user extensibility for data types, operators and functions
 - The same type/function catalogue allows users to create their own functions and operators on the fly.
- Provide facilities for active databases (alerts, triggers) and inferencing, including forwards- and backwards-chaining
 - Triggers were a new idea then. Alerts, which are still a feature of PostgreSQL, are still not a widely implemented concept. (You can have a client “listen” on a database connection for alerts, and take actions, without continuous database polling.)
- Simplify the DBMS code for crash recovery
 - Architectural improvement over Ingres.
- Produce a design that can take advantage of optical disks and VLSI chip designs
 - A reference to the “hot cool” technologies of the mid-1980s. VLSI = “very large scale integration”, basically refers to fast complex chips, like we have today.
- Make as few changes as possible, preferably none, to the relational model

WWW.REFRACTIONS.NET

What is PostgreSQL?

- **1986-1994: Postgres**
 - Original research project
- **1995: Postgres95**
 - SQL support
- **1996-2000: PostgreSQL 6.0-7.0**
 - Open source team
 - MVCC, stability, performance
- **2001-2005: PostgreSQL 7.0-8.0**
 - SQL92, complex SQL, schemas, optimizer, Win32
- **2006-2007: PostgreSQL 8.1, 8.2**
 - In-memory bitmap, roles, partitioning, performance

• Postgres spent about 10 years as a research project, with its own query dialect, “PostQUEL”. It was a good enough database to have a large community of academic and other users who wanted to see it continue being developed after the end of the research project phase.

• Postgres95 was put out by graduate students of Stonebraker, Andrew Yu and Jolly Chen.

• Members of the Postgres user community took up the development after Yu and Chen moved on to other pursuits. These users became the PostgreSQL global development community.


• The first development cycle involved the new development team understanding the code base, and working through a large backlog of known performance and stability issues. At the 7.0 release the team felt they “owned” the database, having re-worked and cleaned up every major sub-system by that point.

• At the 8.0 release, Windows support was added, and the user community began to grow beyond the UNIX roots of the database. Development focus has been more “enterprise” features for large installations (roles, partitions, replication, schemas) and ongoing performance and SQL enhancements.

WWW.REFRACTIONS.NET

Why PostgreSQL?

- I need a database!
- *How about MySQL, everyone loves it!*



- I need transactions, triggers, procedural languages, data integrity!

MySQL is “The World’s Most Popular Open Source Database”.

That’s actually their slogan.

Popular for a reason: it is fast at doing the kinds of simple SQL that web developers like.

But it lacks the features of a real database system.

It mangles data inputs it doesn’t understand.

It does not have triggers or a procedural language for server-side programming.

It does not have full transactional guarantees.

But! (I can hear the MySQL proponents chiming in...) It has transactional table types that give you many of those things!

Sure, but to use those types, you have to give up the thing you chose MySQL for in the first place: the speed.

Non-transactional MySQL is faster than every other transactional database.

Transactional MySQL is much slower than every other transactional database.

WWW.REFRACTIONS.NET

Why PostgreSQL?

- *Oh. How about Oracle, everyone buys it!*



- I don't have \$500K.

If someone asked me to build a system on MySQL, I would balk for technical reasons: there are so many places where data integrity is important in a system, do we really want to give that up?

If someone asked me to build a system on Oracle, I would not balk for technical reasons, but I would quite pointedly, are you sure you can afford it? Really, really sure? How far do you think you'll scale it up? Are you really really sure? Are you certainly you won't need the functions you are not paying for? Really, really sure?

Because Oracle's price point ranges from free (as in beer) for XE to unbelievably expensive (for Enterprise) and most real system require the size that XE does not allow and multiple CPUs and failovers, which Enterprise makes very expensive indeed.

WWW.REFRACTIONS.NET

Why PostgreSQL?

- *Oh. How about SQL Server, everyone tolerates it.*

Microsoft®
SQL Server™ 2005

- I don't have \$100K. Besides, it doesn't do spatial (until 2008...?)

SQL Server is the default database of the Microsoft developer set, it's also perfectly good from a technical level, except for being bound to only one deployment platform.


For real deployment, though, it has the same issue as Oracle, just with slightly smaller numbers.

WWW.REFRACTIONS.NET

Why PostgreSQL?

- *Oh. How about PostgreSQL, hardly anyone has heard of it, nobody can pronounce it, those who use it swear by it!*

PostgreSQL



- OK, I'll give it a shot. Does it do spatial?

PostgreSQL does everything MySQL does not, and everything that most users of SQL Server and Oracle need. And it does it at the \$0 price point.

WWW.REFRACTIONS.NET

Why PostgreSQL?

- Feature parity
 - ACID guarantees
 - SQL 92, advanced query optimizer
 - Full text searching
 - Replication
 - Triggers, procedural languages (PL/PgSQL, PL/R)
 - Hot backup, write-ahead logs / PITR
- Better than MySQL
- As good as proprietary
 - Better in some respects

Why not another database? PostgreSQL has all the enterprise features, MySQL does not. PostgreSQL has great spatial, MySQL does not. Proprietary has serious financial scaling problems, and some serious deployment complexity (SQL Server is the exception there). PostgreSQL is functionally as good as proprietary, and better in crucial openness and financial terms.

Why PostgreSQL

- 1997
 - Spend \$50,000 on decent hardware
 - Spend \$50,000 on database software
- 2007
 - Spend \$5,000 on decent hardware
 - Spend \$50,000 on database software
- This is not scalable

WWW.REFRACTIONS.NET

Why PostgreSQL?

Scalability

"Enterprise"	1 Dual-Core	2 Quad-Core
Oracle	\$40,000	\$160,000
IBM DB2	\$36,400	\$145,600
MS SQL Server	\$25,000	\$50,000
IBM Informix	\$50,000	\$200,000
PostgreSQL	\$0	\$0

Vendors like to talk about "scalability" as if customers have infinite money to address their problems. With modern hardware, the main cost driver in scaling up an installation is no longer hardware, it's software. You can buy a dual-quad server with 24Gb of memory from Dell for less than \$10K. Your software licensing for that server will be many times your hardware cost, and that is not a good scalability situation.

Oracle Enterprise Edition pricing 2007-07: $\$40K * NCORES * 0.5$ (for Standard Edition, use \$15K instead of \$40K)

IBM (2007-07) uses "value unit" now, <https://www-112.ibm.com/software/howtobuy/passportadvantage/valueunitcalculator/vucalc.wss>

100 value units for Intel dual core chip, 400 value units for dual Intel quads

DB2 Enterprise = \$364 / value unit

DB2 Workgroup = \$100 / value unit

Informix Enterprise = \$500 / value unit

Microsoft (page updated 2005) SQLServer

<http://www.microsoft.com/sql/howtobuy/default.msp>

Only multiply by processor, not cores! Nice!

<http://www.microsoft.com/sql/howtobuy/multicore.msp>

WWW.REFRACTIONS.NET

What is PostGIS?

- **GEOMETRY**
 - POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINETRING, MULTIPOLYGON, GEOMETRYCOLLECTION, CURVESTRING, CURVEPOLYGON, COMPOUNDCURVE
- **Indexes**
 - R-TREE, linear-time algorithm
- **Functions**
 - OpenGIS "Simple Features for SQL"
 - ISO SQL/MM
 - Over 300 functions

PostGIS is a spatial data type for PostgreSQL. Non-spatial types have functions that do things to them (substring(), trim(), cos(), etc) and indexes to allow fast access (btrees). PostGIS provides the same facility around spatial data. Functions that do things with geometry (Buffer(), Contains(), Union(), Intersects()) and indexes for fast random access (rtrees).

WWW.REFRACTIONS.NET

What is PostGIS?

- Open source
 - General Public License (GPL)
 - Open development and support
- Seven year history
 - 2001: First release, Mapserver support
 - 2002: Improved functions, indexes
 - 2003: GEOS support, many functions
 - 2004: SFSQL conformance
 - 2005: Lightweight geometries
 - 2006: OpenGIS SFSQL compliance
 - 2007: SQL/MM, curves & performance

-2001 was just types and indexes and a couple functions (length, distance, point-in-polygon), enough for data storage and retrieval, and hookup to Mapserver

-2002 added lots of other spatial functions and improved the index implementation

-2003 added the GEOS library and all the spatial relationship functions (contains, intersects, buffer, union) needed for a “real” spatial database

-2004 completed all the functions necessary for OGC support

-2005 changed the on-disk representation to something about 30% smaller for data and 50% smaller for indexes, greatly improving performance

-2006 brought official SFSQL compliance certificate from OGC

-2007 started adding SQL/MM support, more types and more functions, move to ST_* name space for functions

WWW.REFRACTIONS.NET

Why PostGIS?


- Integration
 - Mapserver
 - Geotools (Geoserver, uDig)
 - FDO (Mapguide, Autodesk Map 3D)
 - JUMP (OpenJUMP, Kosmo)
 - OGR (QGIS, Mapserver, GRASS)
 - FME (ArcGIS Data Interoperability Extension)
 - Cadcorp SIS
 - Manifold
 - Ionic Redspider
 - ESRI ArcSDE 9.3
 - Python / Perl / PHP

More and more products support PostGIS as a data source, some products (like Mapguide, Autodesk Map) get support via underlying data libraries.

WWW.REFRACTIONS.NET

Why PostGIS?

- **Simplicity**
 - **PostGIS Polygon**
 - `POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))`
 - **Oracle Polygon**
 - `MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1), MDSYS.SDO_ORDINATE_ARRAY(0,0, 0,1, 1,1, 1,0, 0,0))`



This is the same simple unit square expressed in PostGIS, and in Oracle Spatial.

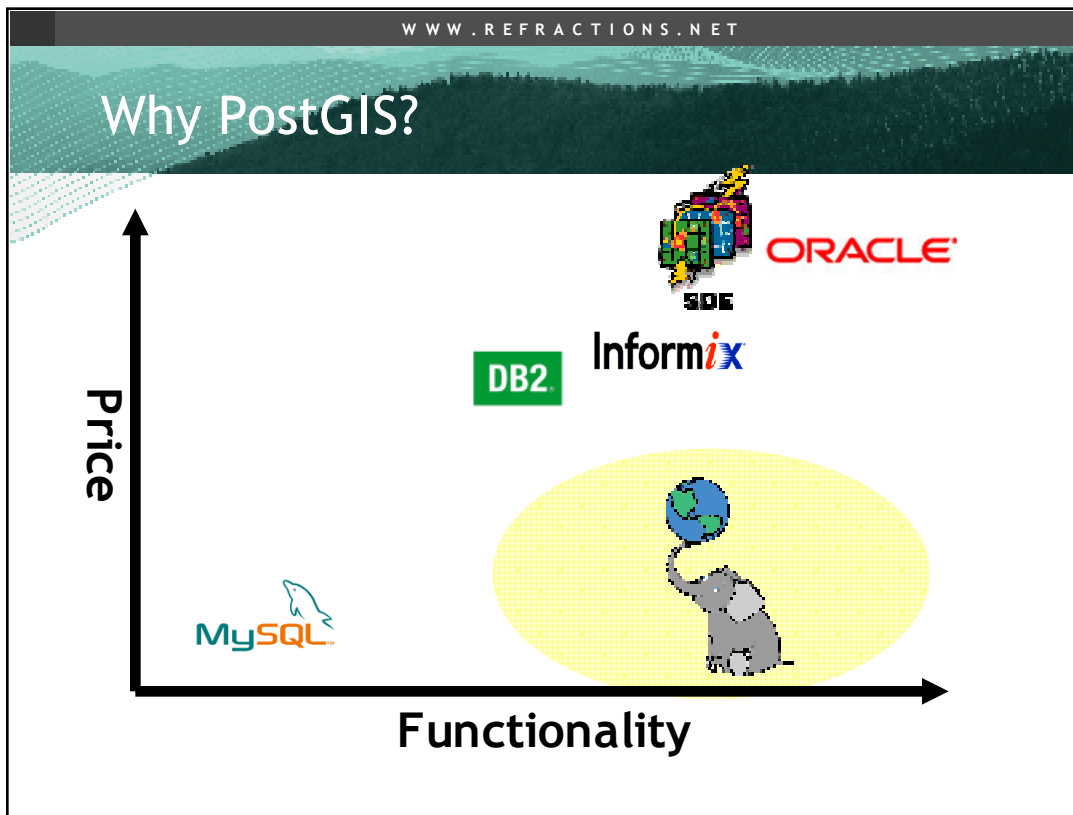
WWW.REFRACTIONS.NET

Why PostGIS?

- **Speed**
 - Lightweight geometry implementation
 - Lightweight indexes (50% size savings)
 - Linear time R-Tree
 - Row-level locking algorithm
- **Features**
 - All OGC Simple Features for SQL, ST_Buffer()
 - Aggregates, ST_Collect(), ST_Union()
 - Extras, ST_AsGML(), ST_AsKML(), ST_AsSVG()
ST_BuildArea(), ST_LineMerge, ST_Transform()

PostGIS uses “lightweight” implementations -- using the minimum amount of bytes to express the geometries and indexes. This reduces the disk size, which is not important because disk is expensive (it is not) but because disk is slow. Really, really, really slow. 100x slower than main memory. The more information you can get off of disk and into memory cache, the faster your system will run. So small indexes are particularly important.

PostGIS has hundreds of functions, all the SFSQL ones, and lots more, that our users have recommended based on their experiences working with the spatial database. Many of these “extra” functions are functions you can only get by buying the most expensive Oracle Enterprise license to get access to full Oracle Spatial.



The main reason to use PostGIS and PostgreSQL is the great location they have in the price/functionalty space. PostGIS/PostgreSQL have lots of features, so they meet the needs of lots of potential users, and they do so at a price point that is far better than the proprietary alternatives. Unless the functionality of the proprietary alternatives is worth the extra cost, why use them?

WWW.REFRACTIONS.NET

Case Studies

- **GlobeXplorer**
 - High performance, good value
- **North Dakota State Water Commission**
 - Good value, legacy integration
- **Institut Geographique National**
 - Good value, better functionality
- **Fleet Management Company**
 - Better performance, same price

The organizations in these case studies all had different reasons to use PostGIS/PostgreSQL. Some actually changed from other databases. Some were putting in their first database. Some were just moving to spatial for the first time. All of them moved **UP** the value chain, increasing the business value of their software by using PostGIS.

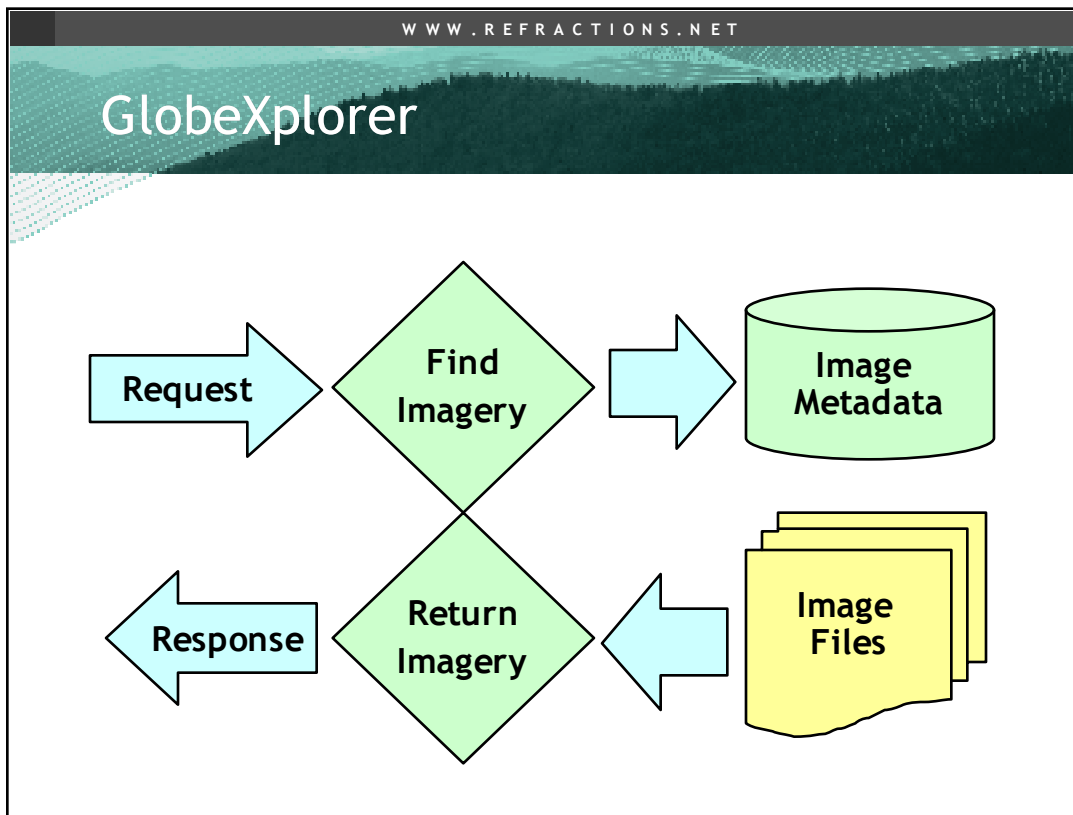
WWW.REFRACTIONS.NET

GlobeXplorer

- www.globexplorer.com
- Consumer and end-user oriented image distribution service (1M-5M requests / day)
- Manage archive of images (multi-terabyte) and vectors (32M parcels and growing)
- Provide access via web services
 - WMS
 - Own web services
 - Desktop extensions



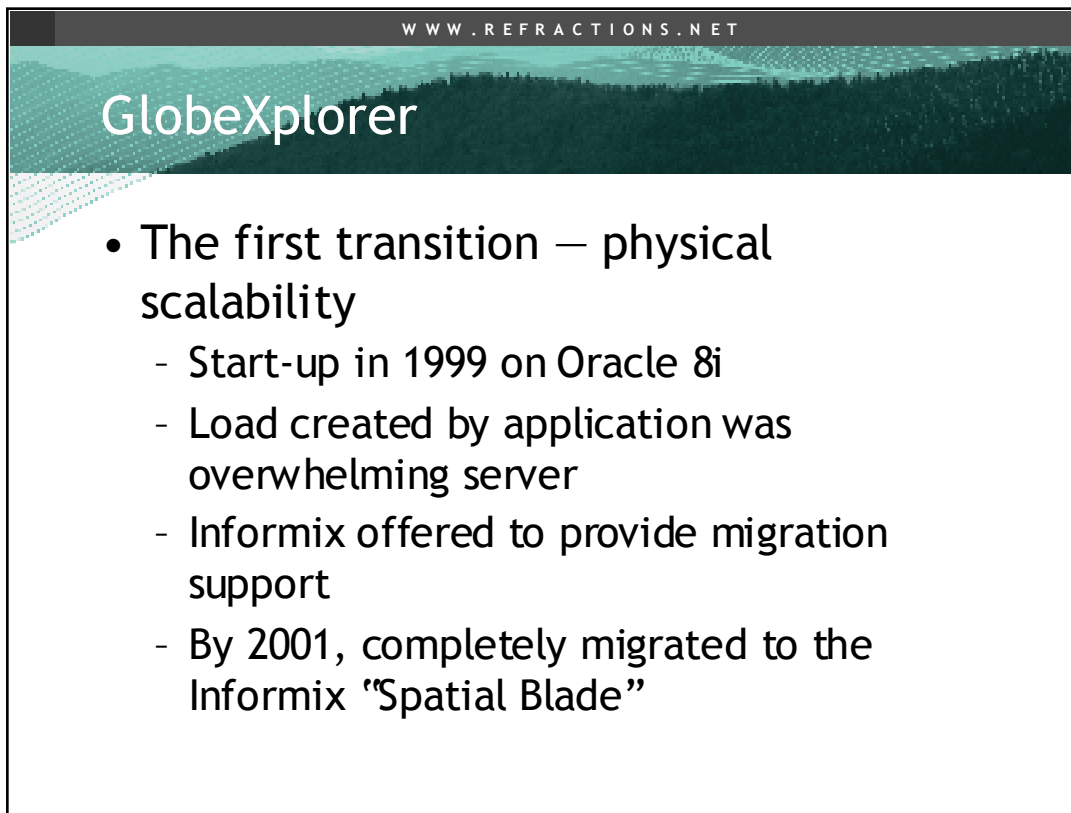
GlobeXplorer is a division of Digital Globe (acquired fairly recently), devoted to pushing imagery out to consumers. Service is based on metered access to data, with most of the money going to the image provider and GX taking a small transaction fee for managing the process. The archive is huge, and the customer base is always growing.



The GX system is a simple web-services application:

- A client makes a request for an image over a certain area.
- The system looks up where the image files for that area are, using a metadata database (also looks up who the image belongs to, and records the access for later billing purposes).
- The system pulls the image files, processes them lightly and sends them back to the client.

The metadata database gets bigger and bigger. It has also moved beyond metadata to include holdings of vector data as well.

The slide features a dark green header with a city skyline silhouette and the text 'WWW.REFRACTIONS.NET'. Below the header, the title 'GlobeXplorer' is displayed in white. The main content area is white and contains a bulleted list of facts about the company's early database migration.

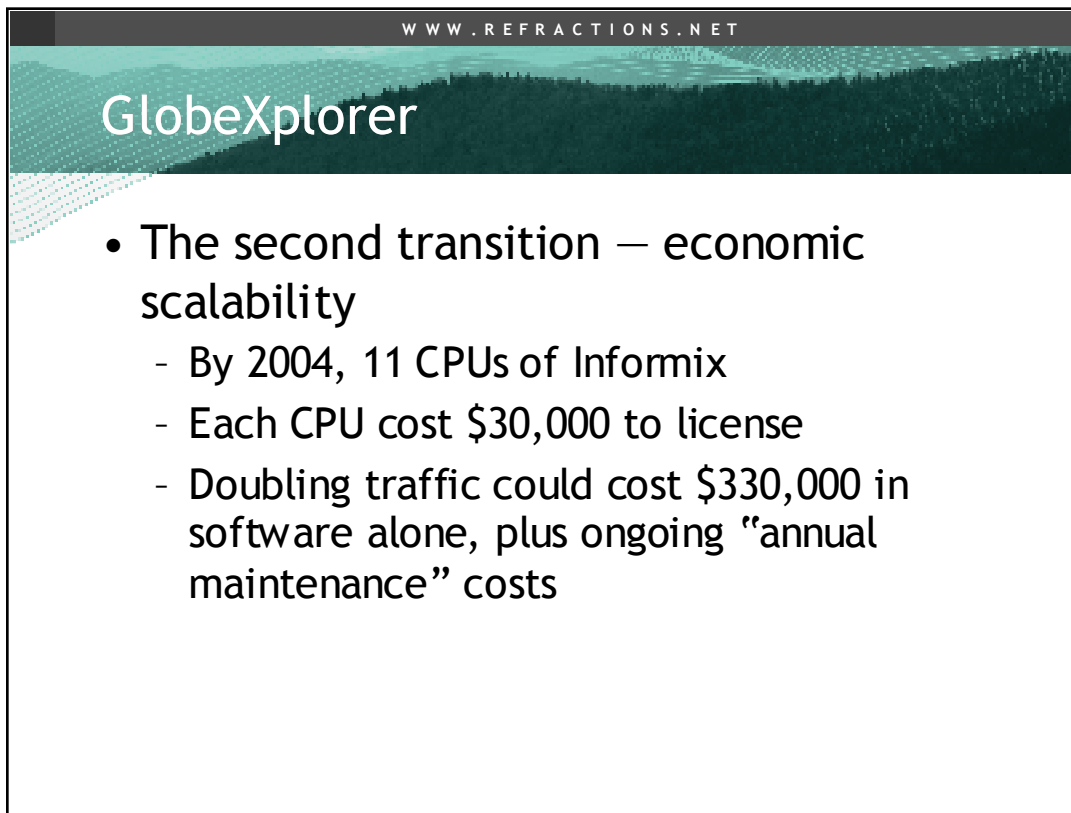
WWW.REFRACTIONS.NET

GlobeXplorer

- The first transition – physical scalability
 - Start-up in 1999 on Oracle 8i
 - Load created by application was overwhelming server
 - Informix offered to provide migration support
 - By 2001, completely migrated to the Informix “Spatial Blade”

When the company was first starting up, they experimented with different technologies to meet their needs. Their initial implementation based on Oracle could not scale physically -- it just did not keep up with the load generated by the image-metadata-lookup application.

Informix migration fixed the physical scaling problem, the application could now grow to meet the load, but there was a new problem...

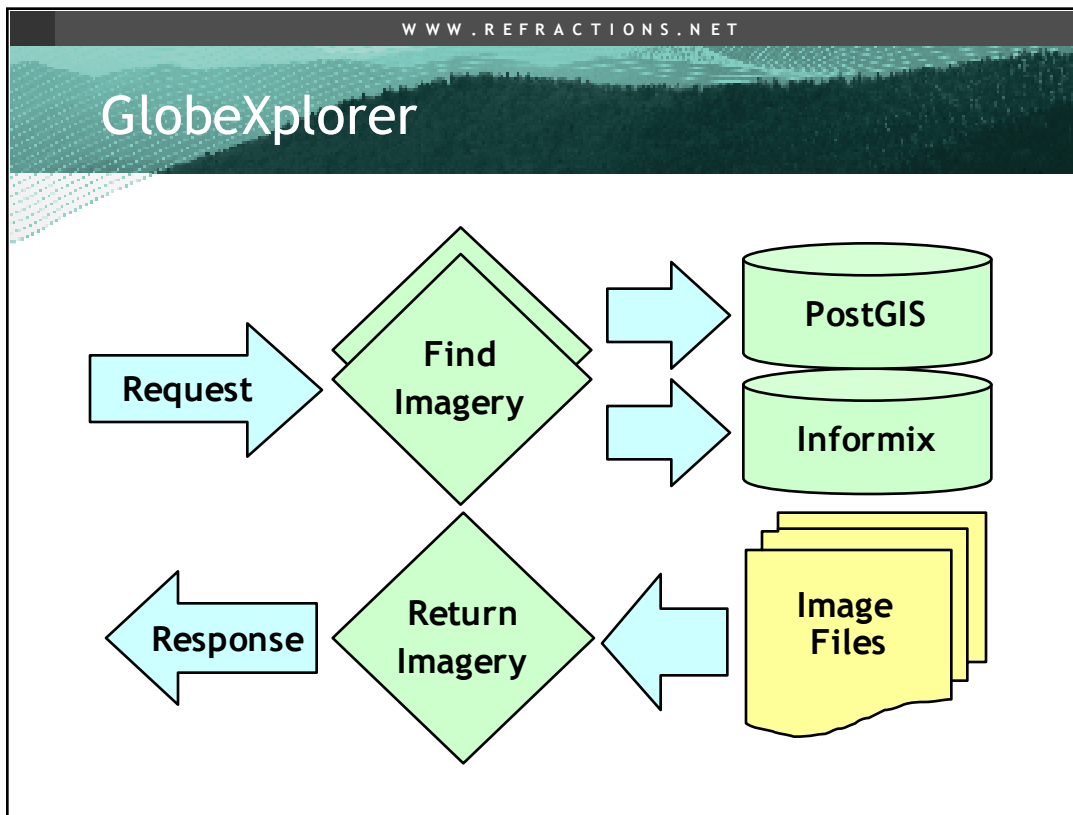


The screenshot shows the top of a website with a dark green header containing the URL "WWW.REFRACTIONS.NET" and the title "GlobeXplorer". Below the header is a white content area with a bulleted list of information.

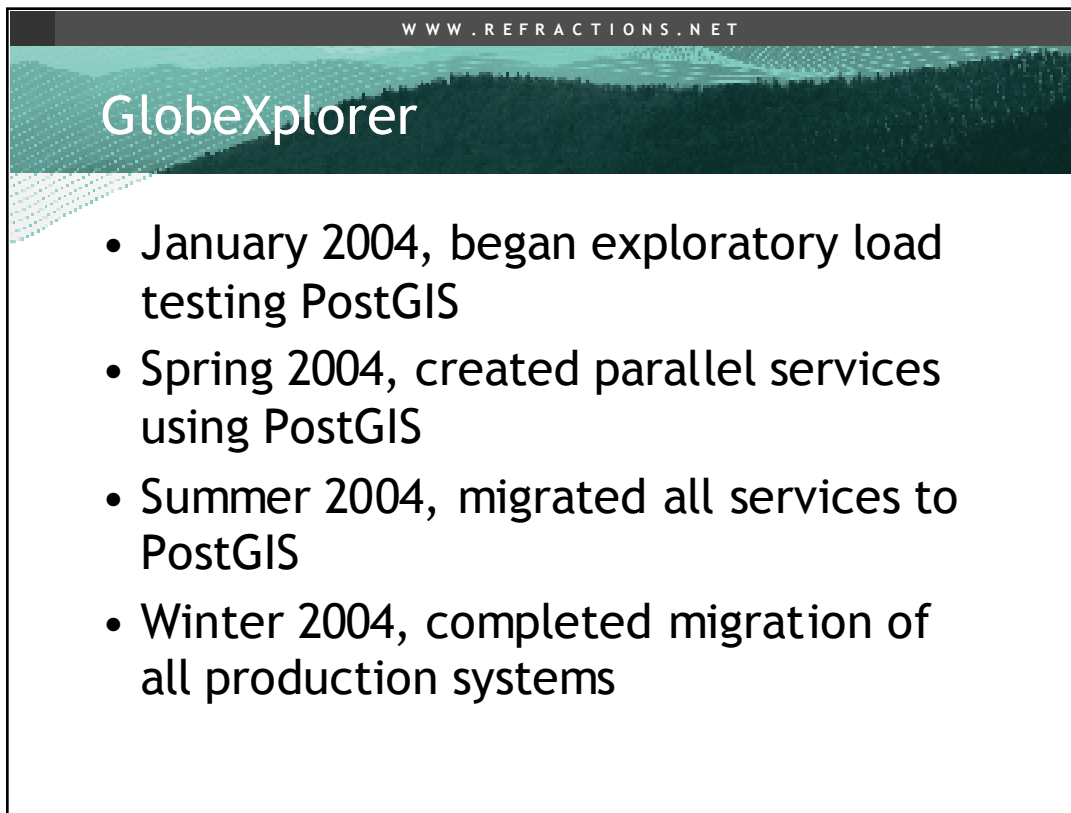
- The second transition – economic scalability
 - By 2004, 11 CPUs of Informix
 - Each CPU cost \$30,000 to license
 - Doubling traffic could cost \$330,000 in software alone, plus ongoing “annual maintenance” costs

The Informix solution was not very financially scalable. New nodes were costing buckets of money, and the prediction for more load meant a huge future outlay was going to be needed.

GX began exploring solutions, and hit on PostGIS. They conducted extensive testing first, to make sure the database would handle their loads.



GX engineers build a parallel system that used a PostGIS backend instead of Informix and ran the full load of the GX system through it, to see how it performed. It could handle the load!

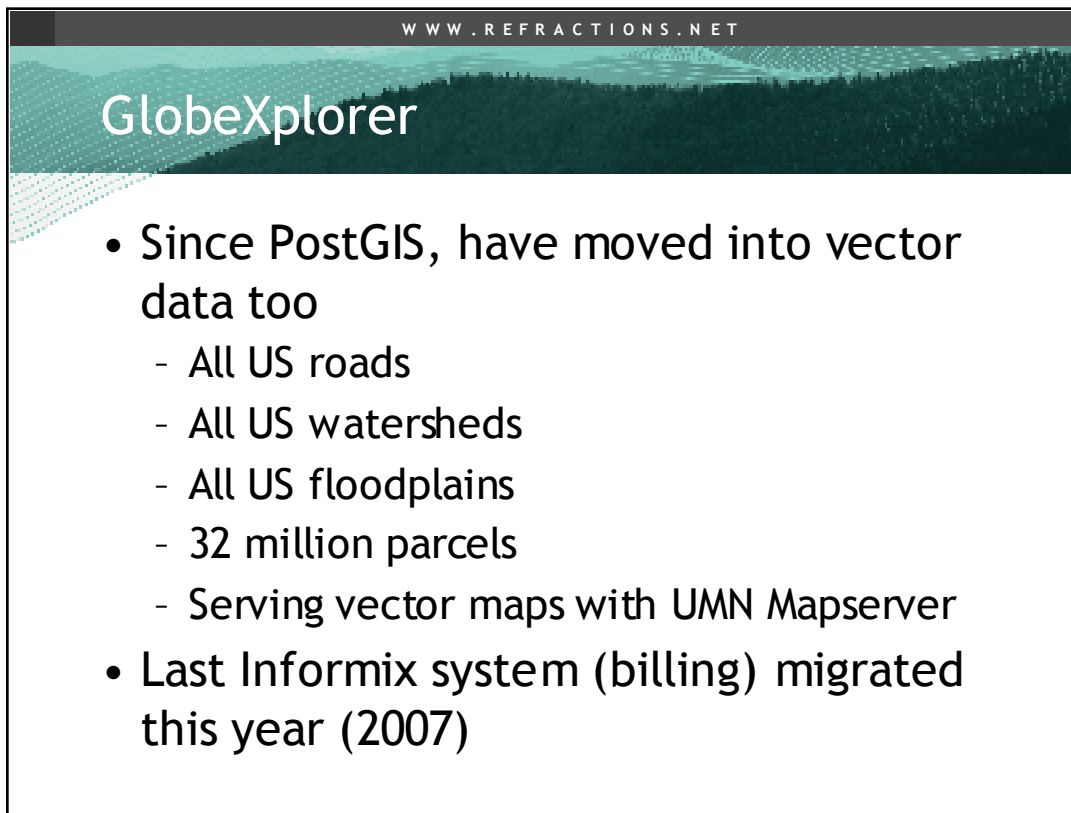


WWW.REFRACTIONS.NET

GlobeXplorer

- January 2004, began exploratory load testing PostGIS
- Spring 2004, created parallel services using PostGIS
- Summer 2004, migrated all services to PostGIS
- Winter 2004, completed migration of all production systems

GX began moving their infrastructure from Informix to PostGIS.

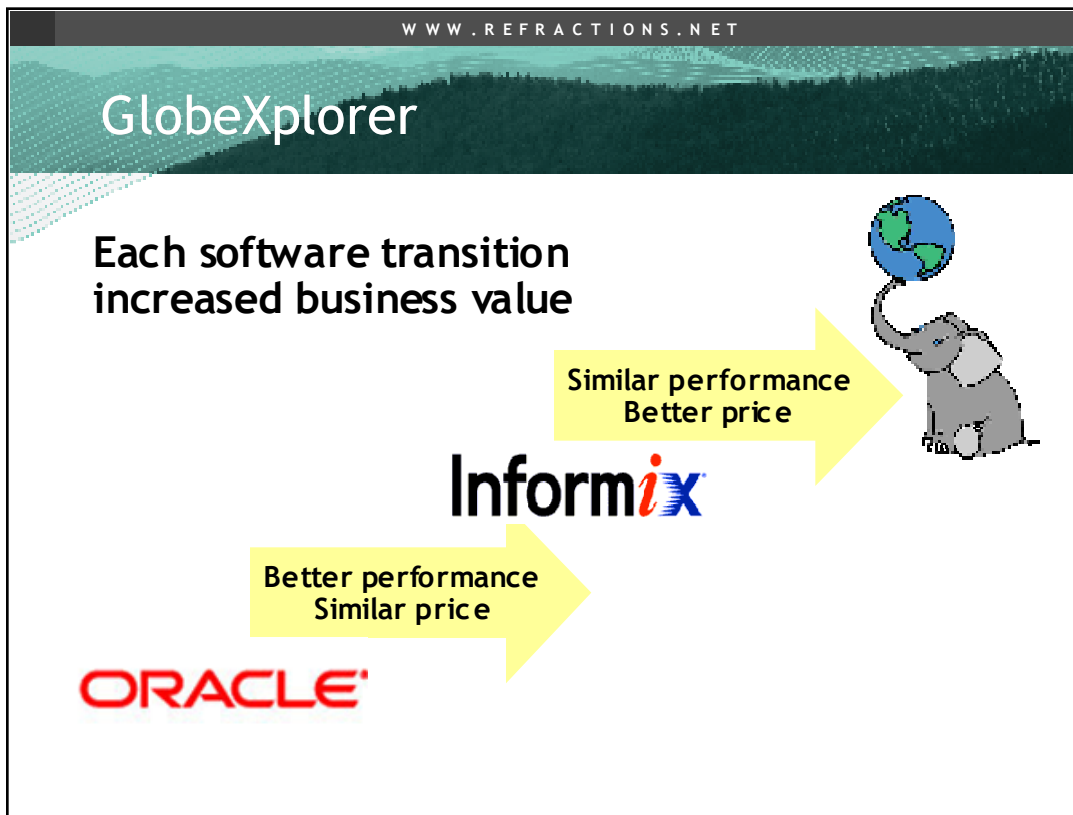
A presentation slide for GlobeXplorer. The top header features the URL 'WWW.REFRACTIONS.NET' in white text on a dark green background. Below the header, the title 'GlobeXplorer' is displayed in a large, white, sans-serif font. The main content area is white and contains a bulleted list of key features and milestones. The background of the slide features a stylized, dark green landscape with a grid pattern.

WWW.REFRACTIONS.NET

GlobeXplorer

- Since PostGIS, have moved into vector data too
 - All US roads
 - All US watersheds
 - All US floodplains
 - 32 million parcels
 - Serving vector maps with UMN Mapserver
- Last Informix system (billing) migrated this year (2007)

GX has also moved into managing access to vector data, and loaded that information into PostGIS. Mapserver is used to generate the cartographic output.



Notice how each transition moved GX to a higher point in the performance /price graph... Oracle offered insufficient performance at a high price. Informix offered better performance at a high price. PostGIS offered the same performance at a better price.



North Dakota is a state in the center of the USA, in the midst of the Great Plains. As a primarily farming state, the allocation of water is a politically sensitive business: there is relatively little of it, and it is in great demand.

WWW.REFRACTIONS.NET

North Dakota Water Commission

- Regulates use of water resources
- Employs scientific experts (hydrologists) and decision makers
- Monitor water levels, water quality, and water usage
 - Generate a great deal of data
 - Water meters, water samples, etc

Water commission is in charge of producing information about water use, and regulating the use of water. The water system, rivers and wells, are extensively metered, producing a large quantity of hydrological data.

WWW.REFRACTIONS.NET

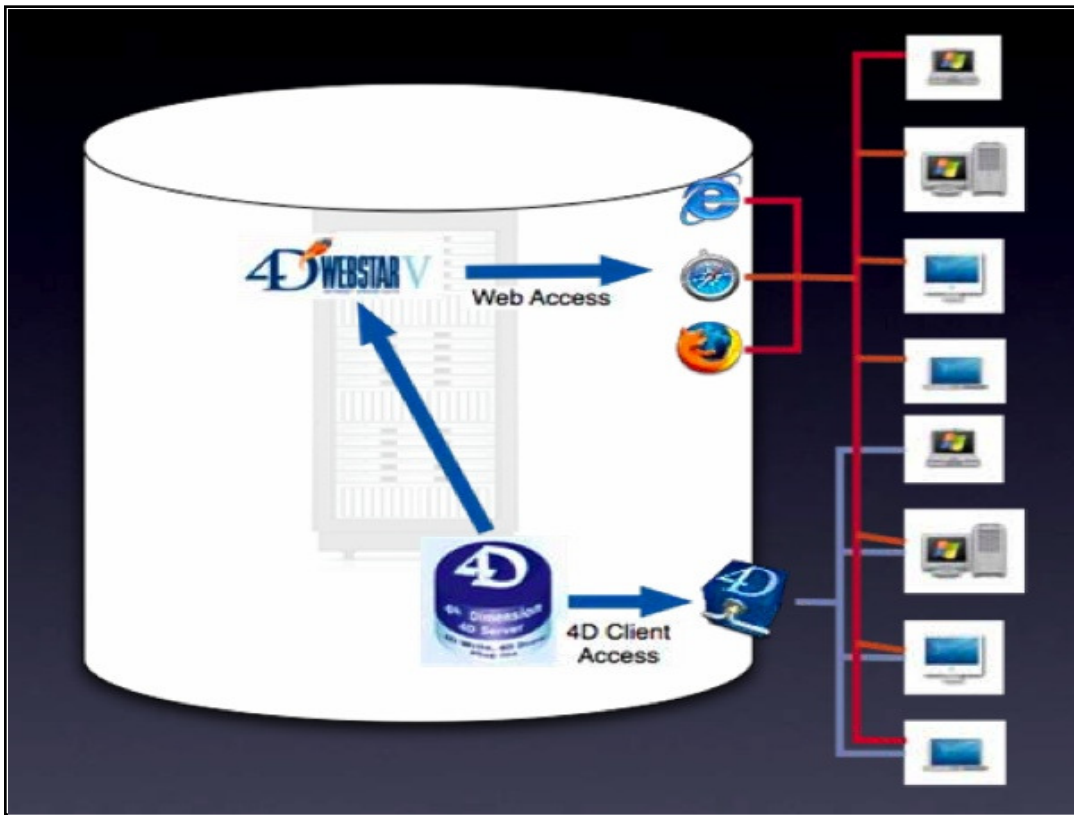
North Dakota Water Commission

- 2001 plan to migrate to ESRI
 - ArcIMS, ArcSDE, ArcMap
- 2003 state budget cut back
 - Migration stopped
 - \$500,000 less than originally planned
- New options needed
 - Mapserver
 - PostGIS / PostgreSQL

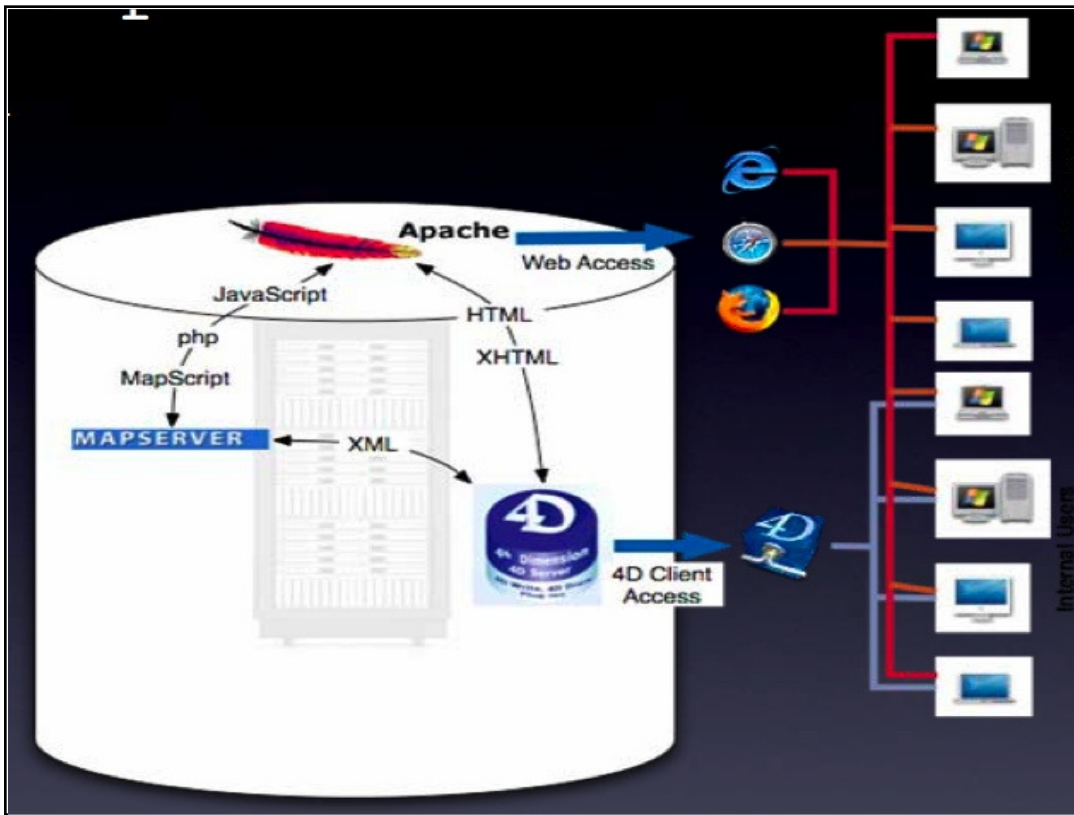
The legacy system in the Water Commission was built in the “4D” database on Apple Macintosh computers (a Windows analogue might be FoxPro). In 2001, the commission planned to start providing more spatial mapping and analysis by migrating to a new spatial data system based on ESRI software. However, budget cutbacks drastically reduced the amount of money available for capital expenditures. Staff began exploring alternatives.



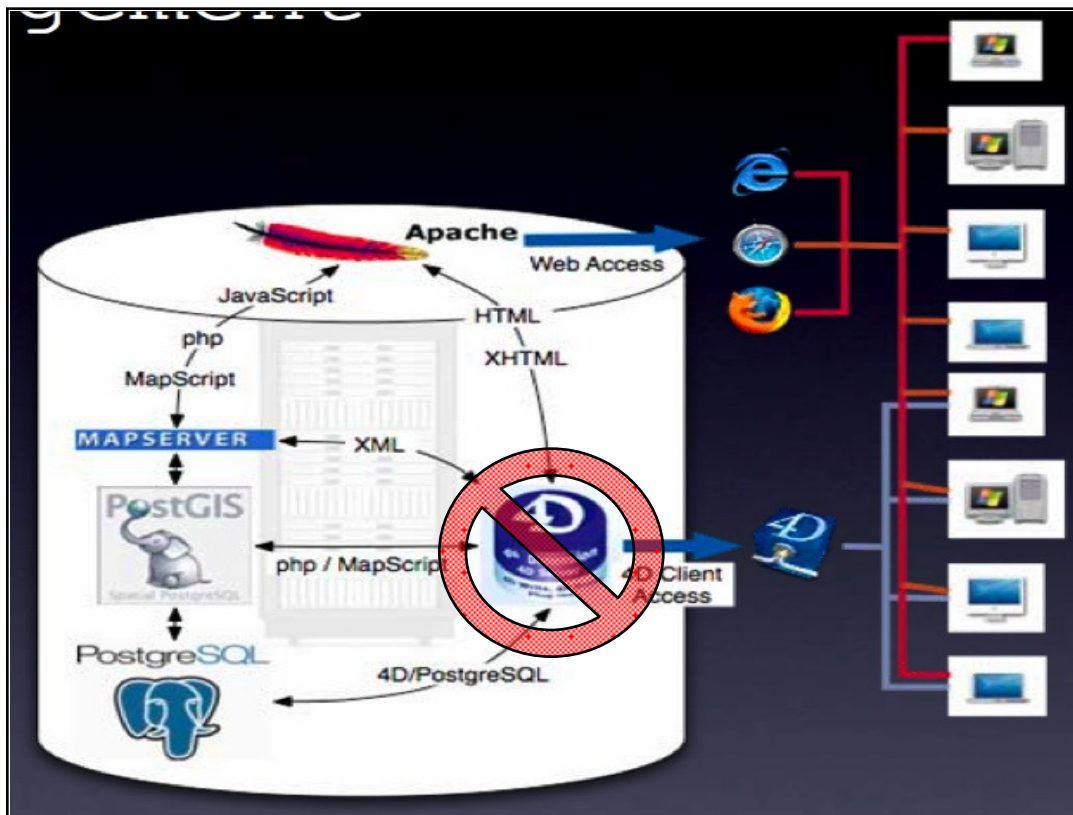
Original system was thick client software on PCs and a 4D server providing just attribute data.



They added web access to that data with a web extension from the 4D vendor.



Then they wanted to link the attributes to maps, but 4D had nothing for that, so they replaced the 4D web extension with Apache/PHP and started using Mapserver to generate their web maps.

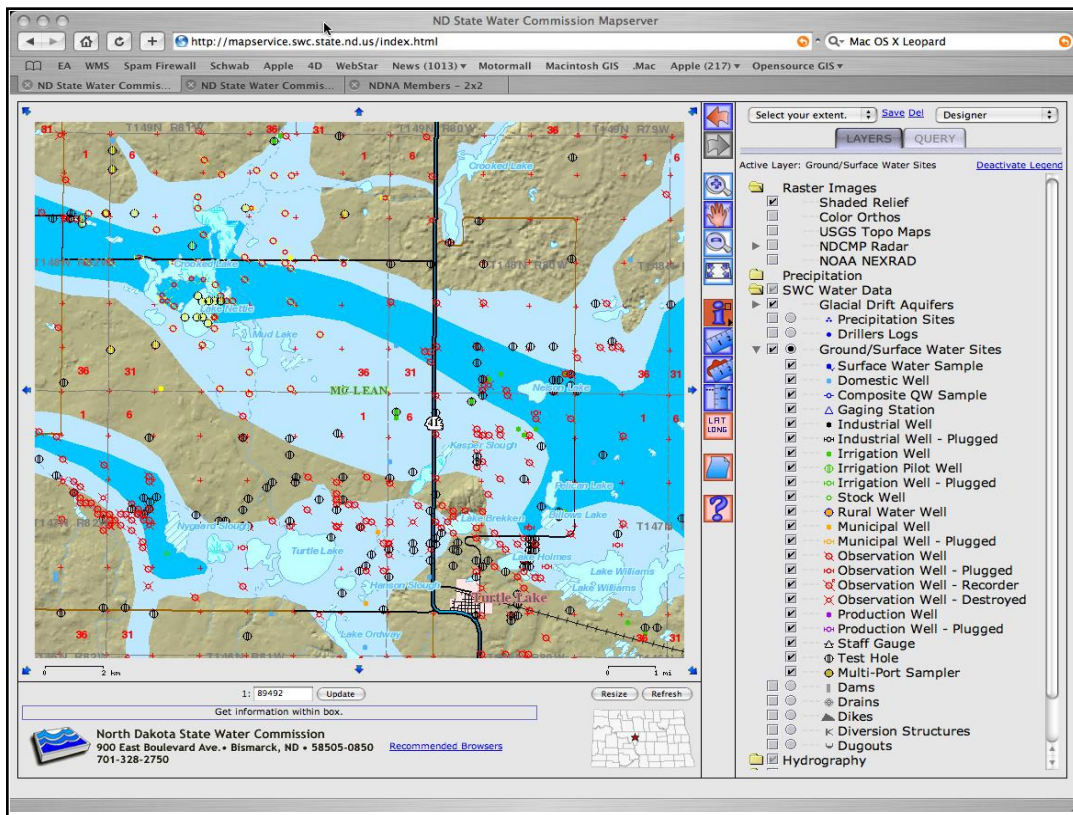


The system they built used PostGIS and Mapserver, while maintaining the legacy database applications, to avoid a complete re-build of their systems. The hydrologic data in the 4D database is still managed and accessed by scientists using the 4D clients. Data is extracted from 4D with ODBC, and piped into PostgreSQL, where it is joined to spatial entities in PostGIS, and pushed out to maps via Mapserver. This system allows them to continue to serve their internal needs while adding new external clients over the web.

The final stage will be the decommissioning of 4D and exclusive use of PostgreSQL as their main database server.

They already have scientists in the commission hitting the PostgreSQL database directly with scripts that bundle the raw data for import into hydrological model software, because connectivity with PostgreSQL is more standard than with 4D.

PostGIS Introduction and Case Studies



This is the map of wells and plots, created using Mapserver and PostGIS.

WWW.REFRACTIONS.NET

North Dakota Water Commission

- Data integrated
 - 2,000,000 water level records
 - 31,000 well sites
 - 54,000 chemistry analyses
- Bonuses
 - Open source was easier to install and maintain
 - Spatial / attribute integration opening up analysis possibilities for hydrologists

Moving to PostGIS has allowed the IT folks to start to provide direct access to scientists who are extracting and analyzing data directly from the database using scripting languages to pull out observations and feed them into hydrologic models. The plan is to eventually shut down the 4D database and manage all the data in PostgreSQL / PostGIS exclusively.

WWW.REFRACTIONS.NET

North Dakota Water Commission


The diagram illustrates a software stack migration. On the left, the ESRI stack is represented by the ArcIMS logo (a globe with yellow lines) and the SDE logo (a stack of colorful cubes). A large yellow arrow points from this stack to the right, containing the text: "More flexibility", "Better manageability", and "Better price". On the right, the open source stack is represented by the MapServer logo (the word "MapServer" with a globe icon) and the PostGIS logo (the word "PostGIS" with an elephant icon and the text "Spatial PostgreSQL" below it).

By replacing an ESRI software stack with an open source Mapserver / PostGIS stack, the Commission was able to move up the functionality/price graph, swapping a proprietary stack for an open source one that provided similar core functionality and better flexibility at a much better price.

WWW.REFRACTIONS.NET

Institut Géographique National

- www.ign.fr
- National mapping agency of France
- Manage topographic database of 100M features
- Provide multiple data collectors read/write access to the central archive




<http://postgis.refrations.net/documentation/casestudies/ign/>

The [Institut Géographique National](http://www.ign.fr) (IGN) is the national mapping agency for France, with 1800 employees and a mandate to collect, integrate, manage and distribute reference geographical information for the whole country.

WWW.REFRACTIONS.NET

Institut Géographique National

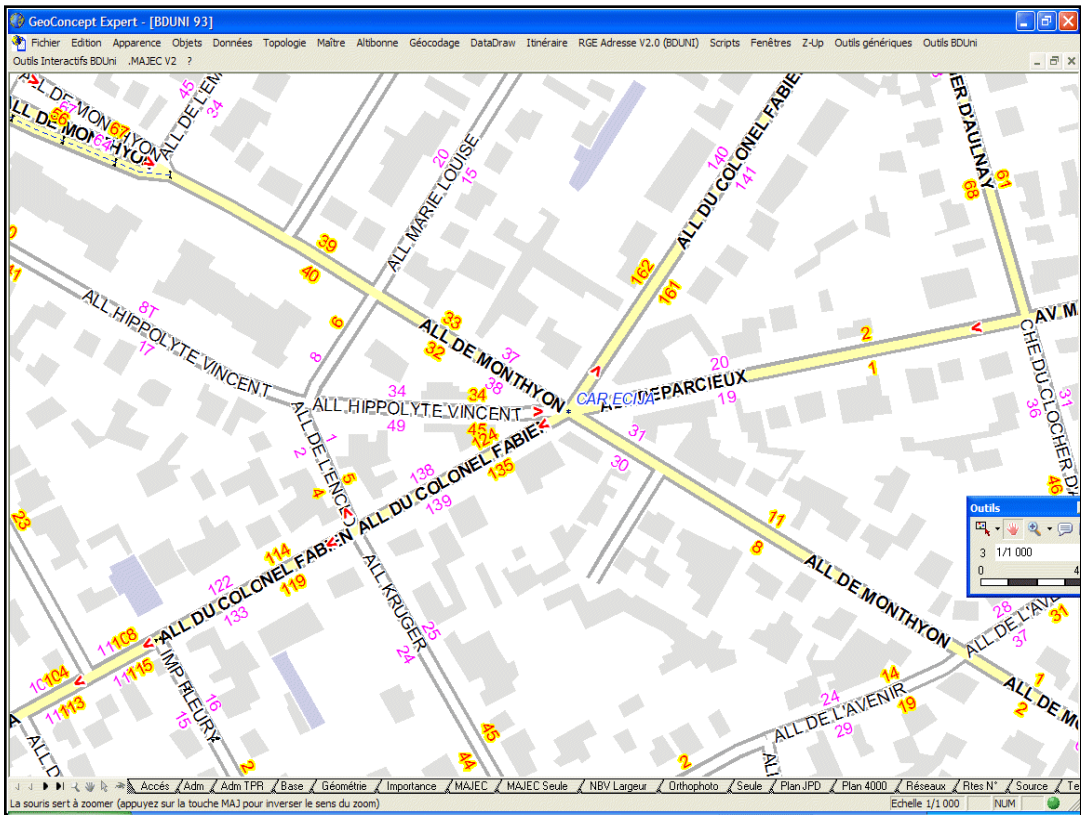
- BDUni geodatabase
 - 1:25000 topographic maps
 - 3D roads, road names, bridges, hydrography, buildings, administrative boundaries
- Managed with desktop GIS software (GeoConcept) 
- 120 field researchers based in six cities
- Flat files organized into tiles

IGN built and managed Bduni originally with the GeoConcept desktop GIS software. GeoConcept is a French GIS company. The data volume issue of handling such a large data set was resolved by breaking the national coverage into thousands of smaller map tiles.

Bduni is not a static data set, it is under constant revision and update. As such it needs a management system capable of handling multiple readers and writers, and a data model that can track revisions over time. The static file system worked OK for breaking up read/write concurrency, but lacked a good revision system, or easy access to the entire data set for query purposes.

IGN began looking into a spatial database, so they could store the entire data set in one seamless database.

PostGIS Introduction and Case Studies



This is an example of Bduni data, as seen in the GeoConcept interface.

WWW.REFRACTIONS.NET

Institut Géographique National

- Reviewed database alternatives
 - Oracle Spatial
 - DB2 Spatial
 - PostGIS / PostgreSQL
- Questions of interest
 - Capable of handling 100M record database
 - Reasonable speed with such a size
 - Full transactional integrity for spatial objects
- All databases were acceptable
 - PostGIS had the best performance/price ratio (∞)
 - No expensive tendering process to select PostGIS

IGN studied three databases, and determined that they were all capable of handling their anticipated database size. However, PostGIS had the best functionality/price combination, and as an added bonus did not require a procurement process prior to deployment -- IGN could get to work with PostGIS right away!

WWW.REFRACTIONS.NET

Institut Géographique National

- Wrote Check-in/Check-out extension for GeoConcept
 - No longer restricted to tiles
 - Use DB transactions to ensure all-in-or-all-out integrity
 - Disconnected editing

Check-in of completed edits

GeoConcept

PostGIS
Spatial PostgreSQL

Checkout of working area

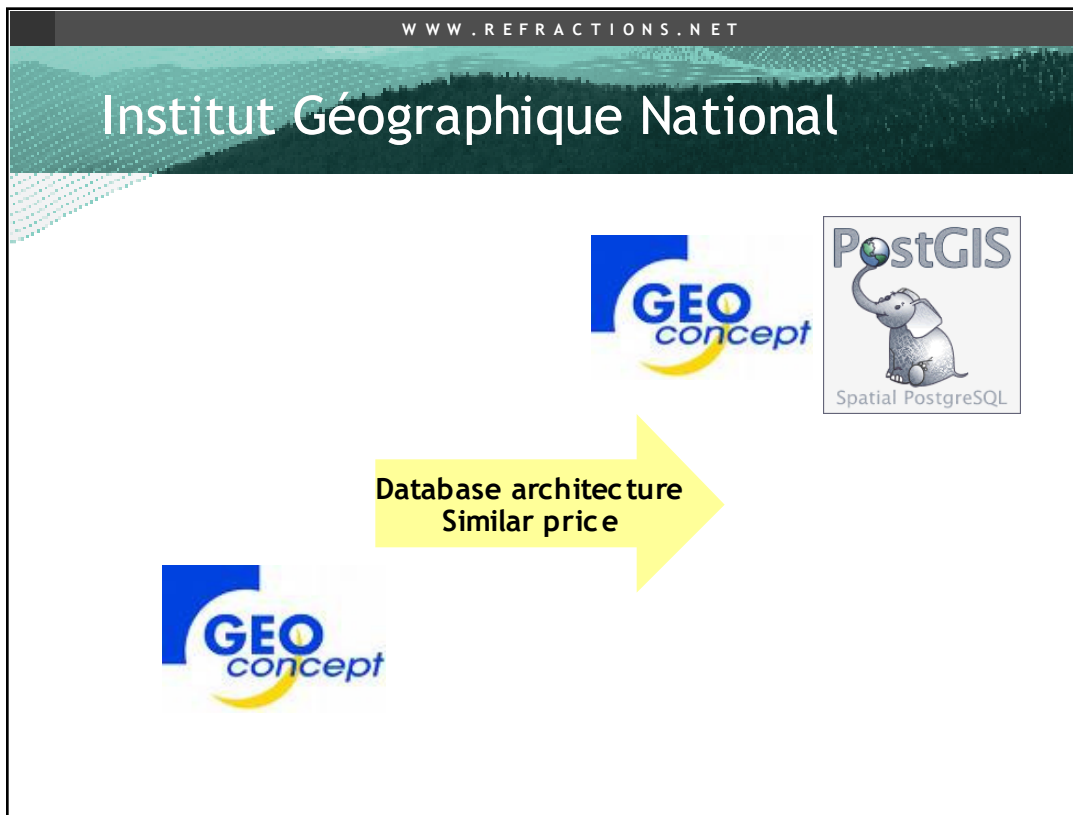
Rather than building a UI from scratch and re-training their operators, IGN retained their existing GeoConcept editing environment, and extended it to allow checkin/checkout from their PostGIS database. The PostGIS database is hooked up to a remote replica, so that it can be kept online at all times. Using GeoConcept allows IGN operators to work disconnected in the field, then synchronize their data when they return to the office. The check-in process allows operators to resolve conflicts between their copy of the data and the central database.

WWW.REFRACTIONS.NET

Institut Géographique National

- First features loaded in July 2006
- All features loaded by end of 2006
- 120 data maintainers connected to the database
- Running with warm standby server replicated in remote office
 - WAL log shipping

Rather than building a UI from scratch and re-training their operators, IGN retained their existing GeoConcept editing environment, and extended it to allow checkin/checkout from their PostGIS database. The PostGIS database is hooked up to a remote replica, so that it can be kept online at all times. Using GeoConcept allows IGN operators to work disconnected in the field, then synchronize their data when they return to the office. The check-in process allows operators to resolve conflicts between their copy of the data and the central database.



IGN moved from a legacy system to one with higher value, without taking on a new high-cost architecture. Compared to a proprietary database / geoconcept combination, the PostGIS architecture also has substantial cost savings.

Fleet Management Company

- Refractions Research client
- Start-up company
 - Selected PostgreSQL early
- Started with fleet maintenance application
 - Download inspection and engine details to central PostgreSQL database daily
 - Popular with school bus fleets, stringent safety tracking requirement

Fleet Management Company

- Added GPS capability to their devices
- Now gathering a GPS sample every six seconds from each vehicle
- Huge volumes of spatial data
 - One fleet = 100 vehicles * 8 hours * 60 minutes * 10 samples / minute = 480,000 samples per day = 120 million samples per year
 - Needed a database and reporting interface
 - Manage, query and display new real time location data

WWW.REFRACTIONS.NET

Fleet Management Company

- Specialized Queries
 - How many times did the vehicle stop? Where? For how long?
 - What mileage did the vehicle cover in Sacramento in June? The whole fleet?
 - What was the maximum speed of the vehicle yesterday? Last week?
 - What route did the vehicle take yesterday? A week ago? A month ago?

Wanted reporting, not just tracking

Not just “show me a map” or “show me where the assets are now?”

Detail!

WWW.REFRACTIONS.NET

Fleet Management Company

- **Specialized Requirements**
 - Huge volumes of data
 - PostGIS lightweight storage
 - Spatio-temporal data
 - PostGIS linear referencing
 - PostgreSQL n-dimensional cube indexes
 - Specialized multi-key queries
 - PostgreSQL GiST multi-key capability
 - PostgreSQL in-memory bitmaps

Hard requirements: massive data, and complex queries.

WWW.REFRACTIONS.NET

Zonar Systems Asset Path Report - Mozilla Firefox

https://sdemo.zonarsystems.net/report.php

Home | Inspections | Drivers | Assets | GPS | Reports | Help

Vehicle Path Report

jump to location: address, city or ZIP: state:

Add assets to the report by selecting them below and clicking 'Generate Report.'

Asset No: (select up to 5)

- GSM_0011
- select one-
- select one-
- select one-
- select one-

Date Range:

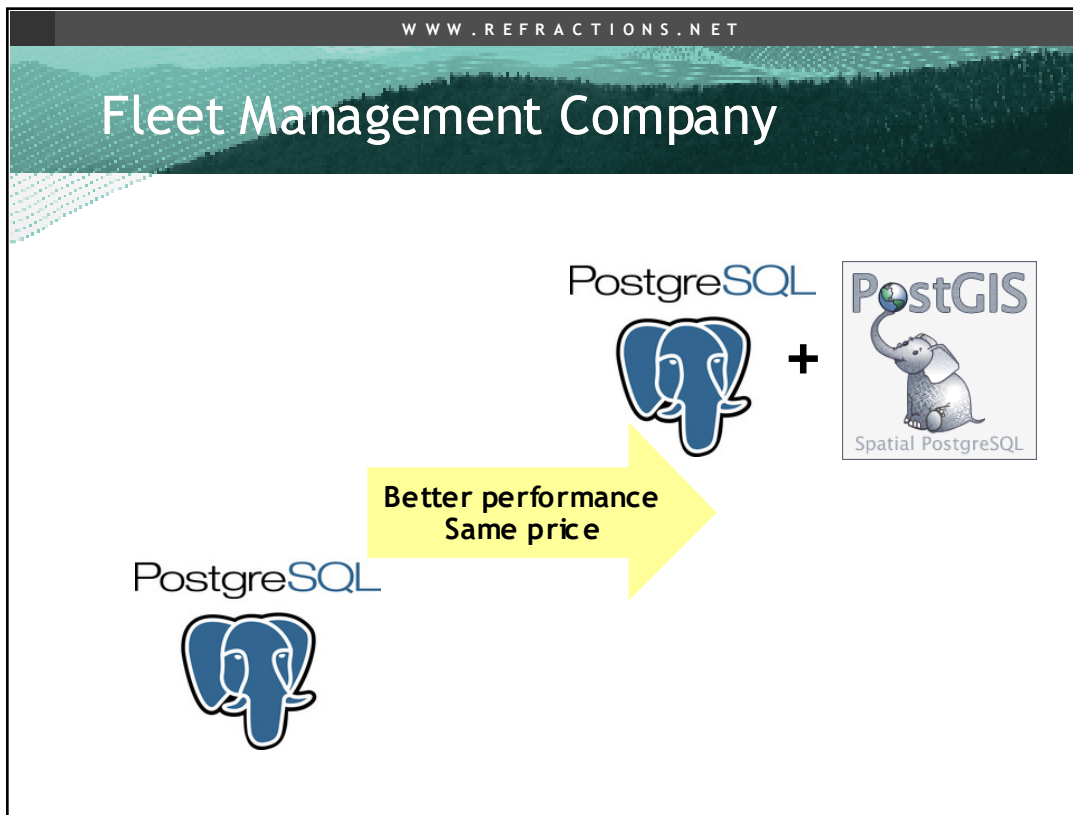
from: 2005 Jul 5 00:00

to: 2005 Jul 7 23:59

Vehicle Path Report Data

Fleet Management Company

- Take GPS point stream and convert to 4-dimensional lines
 - Even smaller data footprint
 - Faster to map than points
- Use Mapserver and Ka-Map to create interactive mapping interface
- Use AJAX methods for all tabular results



WWW.REFRACTIONS.NET

Conclusion

- PostGIS is an essential part of a wide range of spatial systems use cases
 - High performance, simple query
 - Complex integration, legacy systems
 - High performance, complex query
- For many applications, moving to open source will be a move that increases business value
 - Either by lowering costs
 - Or by increasing functionality

We are seeing PostGIS put into deployment in all sorts of environments, academic, business, government. Folks doing so are reaping the benefit of database management systems: uniform access via SQL, data integrity guarantees, large data set handling. They are also reaping the benefit of open source: a responsive community, transparent source code, no software cost for scaling their systems out to more machines. Moving to PostgreSQL/PostGIS is saving folks money and time.

WWW.REFRACTIONS.NET

Questions?



PostGIS
*A spatial database
for the rest of us*