Open Source in British Columbia

Experiences and Strategies

Paul Ramsey
pramsey@cleverelephant.ca

The following are my personal observations and conclusions, and do not represent official positions of GeoBC or the Government of British Columbia.

in May, I talked with staff from GeoBC about experiences with open source.
GeoBC is the successor to the Integrated Land Mangement Bureau, which was the successor to the Ministry of Sustainable Resource Management.
In a nutshell, it's the centralized geospatial data management agency for British Columbia.

Operational

Strategic

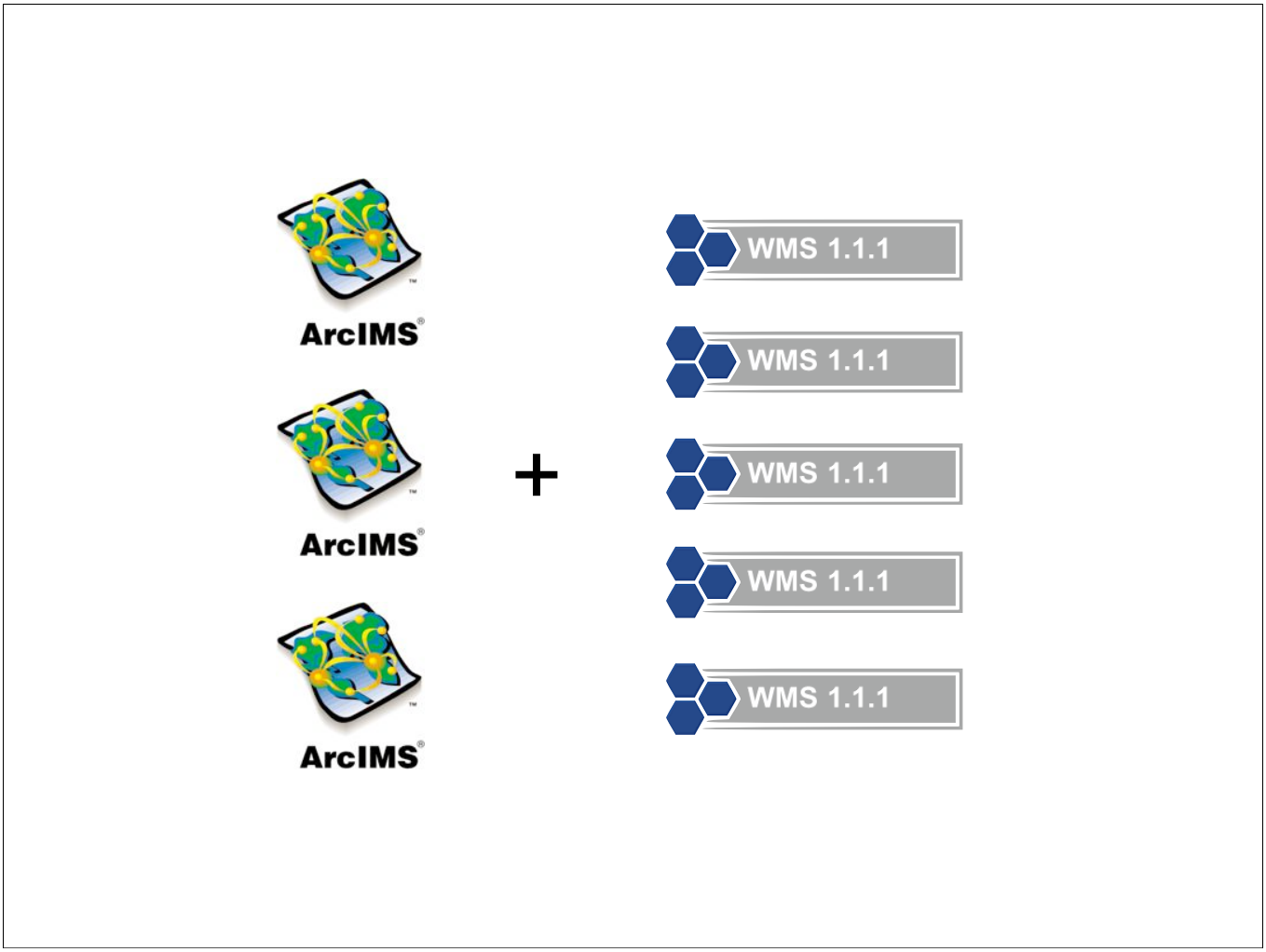
GeoBC has dealt with open source in the context of their operational setup and strategic technology planning.
By no means is GeoBC an open source promoter. Most of their infrastructure is ESRI-based and Oracle stored, and they are committed to maintaining that infrastructure.
But for some particular use cases, they have found pursuing open source fruitful.
They have benefitted, though not always deliberately, from investments in open source.

When I say “operational” I am talking about the day-to-day infrastructure of an organization. Operational decisions are about keeping the wheels turning. Meeting existing needs. Upgrading a database is an operational decision.

Strategic is about planning ahead to allow for new wheels and new needs.
Testing a completely new database technology, like an object database, is a strategic action.
It is preparation for a possible future.
GeoBC, in its various incarnations, has approached open source geospatial both operationally and strategically.

Operational

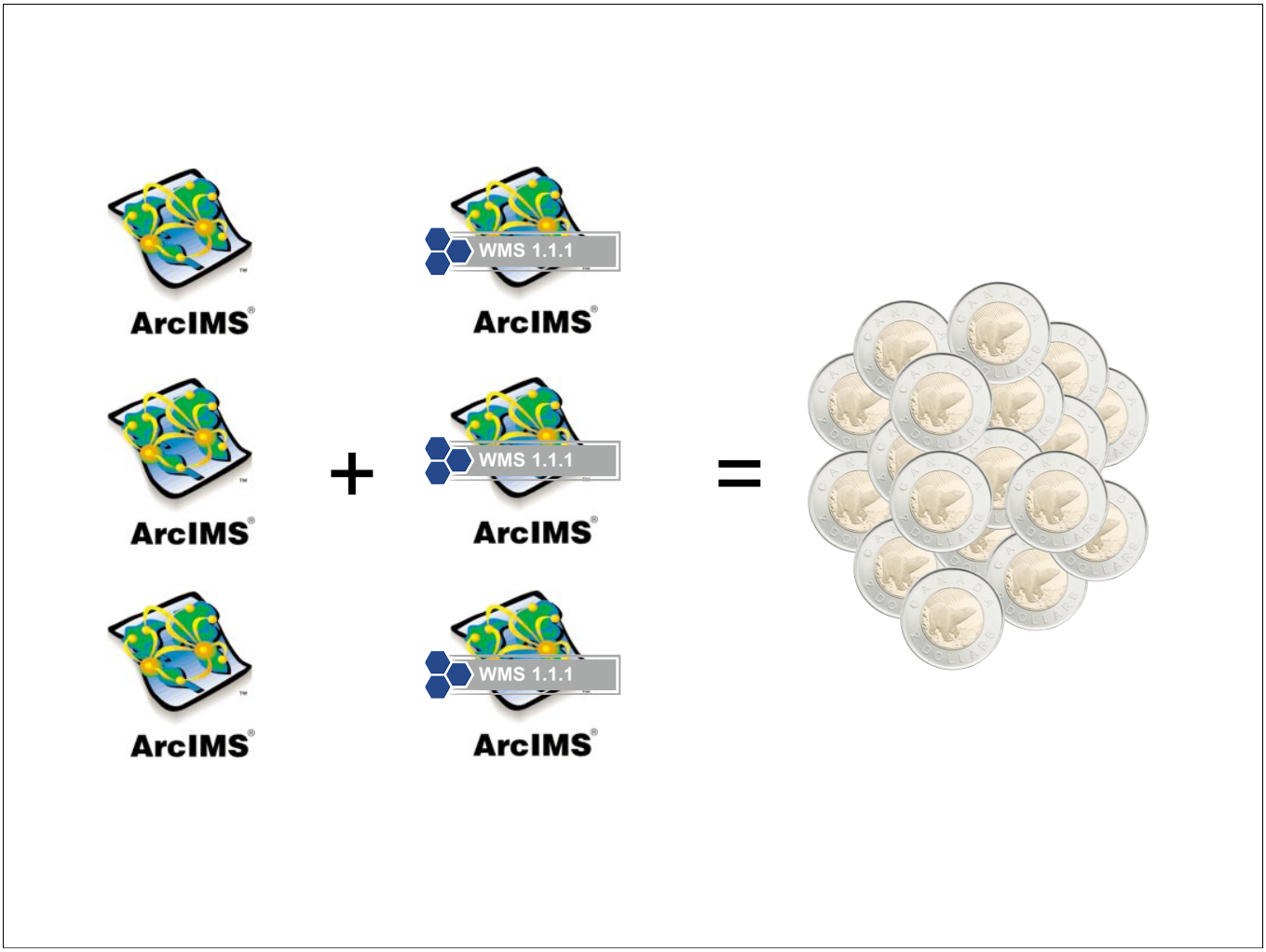
First, the operational side.



The GeoBC web mapping infrastructure was originally all ArcIMS.
As part of a funding mandate from the federation GeoConnections program (more on that later) the province needed to add OGC standard WMS capability to their web mapping servers. They recognized that as a pure machine-to-machine web service, there was a likelihood that the load could grow quite quickly and in an uncontrollable way.

- Scalable
  - Fast + cheap
- Usable
  - Administration + configuration
  - Compatible with ArcSDE
- Standards compliant

So they needed a Web Map Server (WMS) solution that would work with their infrastructure.
It had to scale well, because of the future load.
It had to be easy to administer, because they weren't getting any new staff.
It had to work with the Land and Resource Data Warehouse, a complete compendium of all British Columbia spatial data in an ArcSDE server.
It had to fully support the WMS standard.

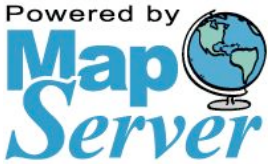


ArcIMS had a “WMS Connector” add-on that would allow ArcIMS services to be published as WMS. But even assuming the scalability was passable physically, the future load could make it a pricey option. They had already done some testing of the Connector and found the administration was not easy. They were prepared to stick with it, because they already had ArcIMS expertise and installations. But the known downsides drove them to look for alternatives.

Mapserver seemed like a good alternative.
It was WMS compliant.
It supported ArcSDE, at least in theory.

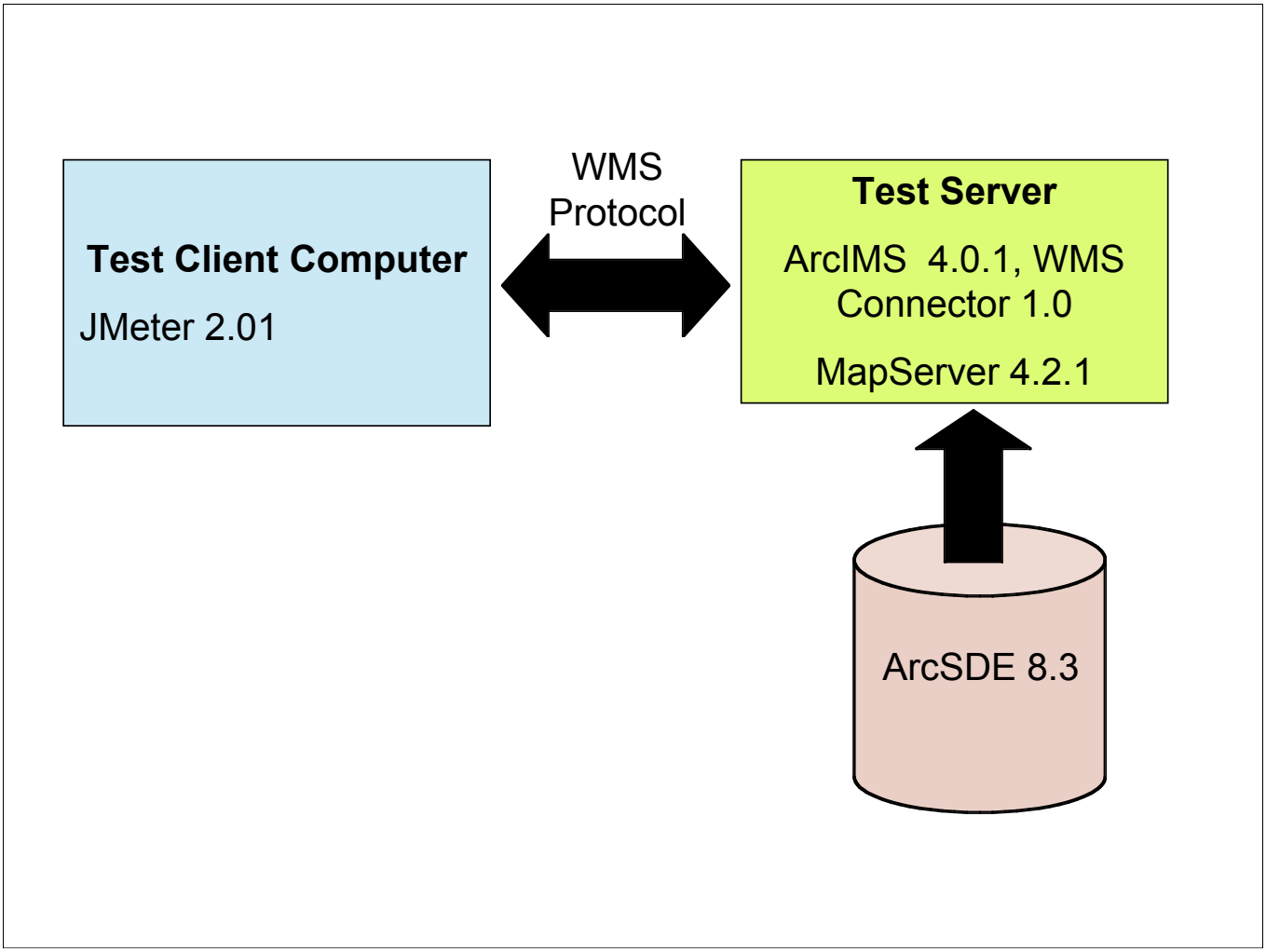
Test the options!

Rather than just read the spec sheets, GeoBC approached the operational problem empirically. Given the two alternatives, which one was going to meet the most criteria most effectively?



- Standard map
- Re-projection
- Lots of features
- Complex features
- Configuration
- Logging
- Image formats
- WMS standards
- Concurrent load
- Administration

They devised a test plan, of over 40 different tests,
- tests of capabilities (“can it do this”)
- tests of performance (“how fast can it do this”)
- tests of usage (“how hard is it to do this”)
The first two were empirically measurable.
The last one involved having a skilled ministry member work through tasks and report on both.
Everything was documented into a complete report.

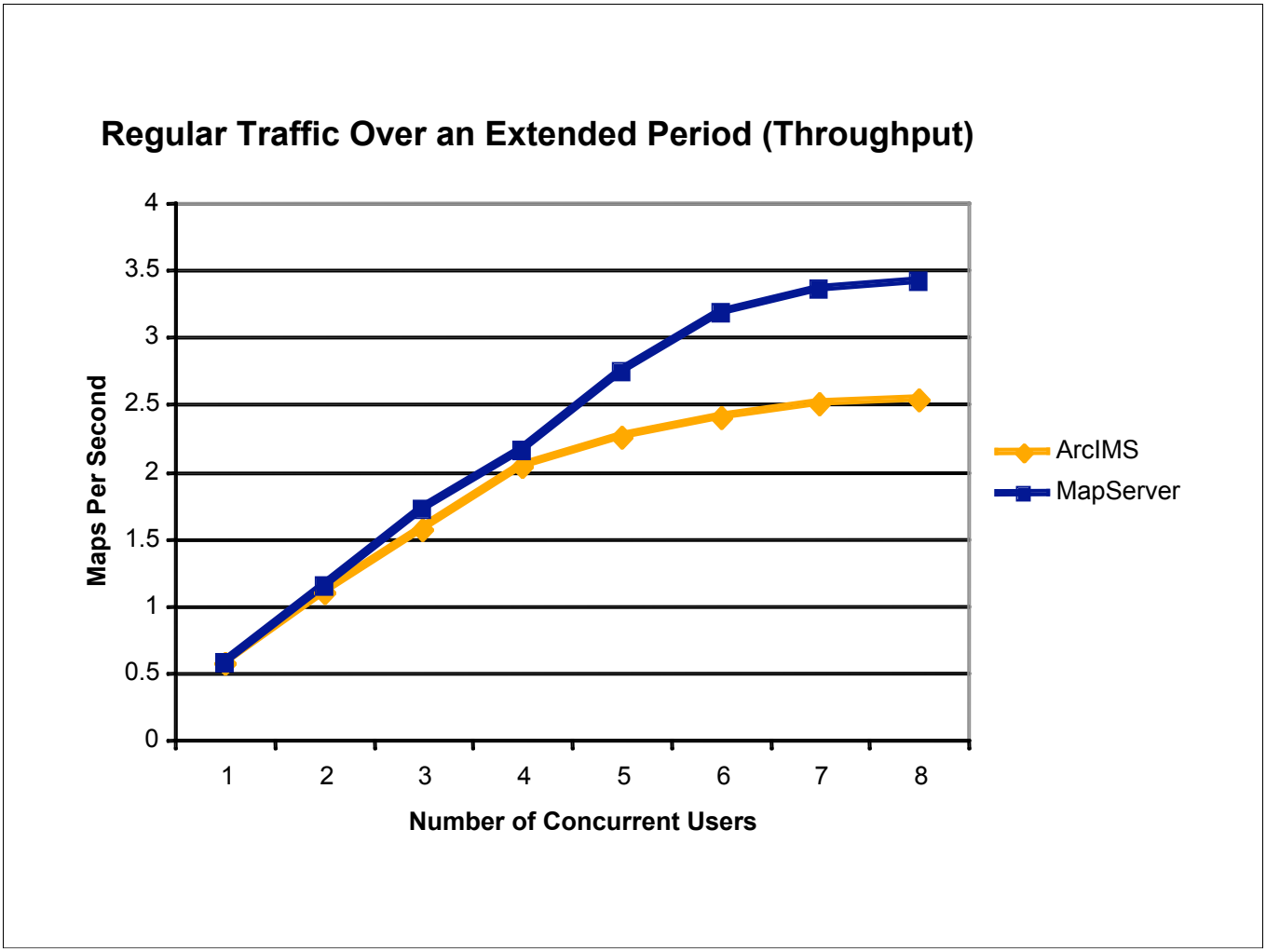


The testing hardness used the infrastructure of GeoBC, and ran both pieces of software on the same box, so there were no hardware differences.

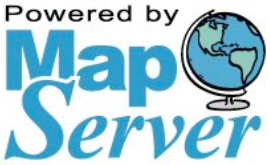
Sucks!

- Too slow!
- Gobbling ArcSDE connections!
- Fix that!
  - FastCGI upgrade (2 weeks)
  - $12,000

The initial results were not good for Mapserver!
In order to test the systems identically, a very large ArcXML configuration had been automatically converted to a Mapserver .map file.
That resulted in a misconfiguration that made Mapserver appear very slow (symbols were rendered as labels, resulting in the label collision algorithm attempting to thin out over 20,000 symbols using an O(N^2) algorithm). It took some expertise to find and remove that error.
With that improvement, things were better, but not great.
It was still quite slow, because it was establishing ArcSDE connections once for each layer.
Fixing that problem required actual code changes, which took two weeks and some extra dollars. But there was budget room in the testing contract, so they went ahead.
With that fix in place, things were much better.



This is just one result of many, but it's the most operational one: how well do the systems handle increasing amounts of concurrent load, basically, how efficiently do they use system resources. The more efficiently they run, the more maps they can squeeze out per second.



Rocks!

- Slightly faster
- More usable
  - Easier administration
  - Easier configuration
- Slightly more standards compliant

The final aggregate results were that in raw performance, Mapserver was somewhat faster. More important, it was much easier to administer, and slightly more standards compliant, though not in an important way. GeoBC decided to use Mapserver for their WMS services.

bespoke

off-the-rack

In conducting their testing, GeoBC recognized a key different of open source, that is possible to make changes to their code in short periods of time, for relatively little money.
In evaluating COTS products, you have to take them “as is”, because vendors will rarely do product upgrades on spec for a single customer, but for open source products that is not necessarily the case.
It might be possible to fix shortcomings in open source products for very small relative amounts of money, and evaluation processes should be aware of that or valuable opportunities could be missed.

Customization	US$12,000
Deployment	US$0
Maintenance	$0 to $5,000

Customization	US$0
Deployment	N * $15,000
Maintenance	N * $3,200

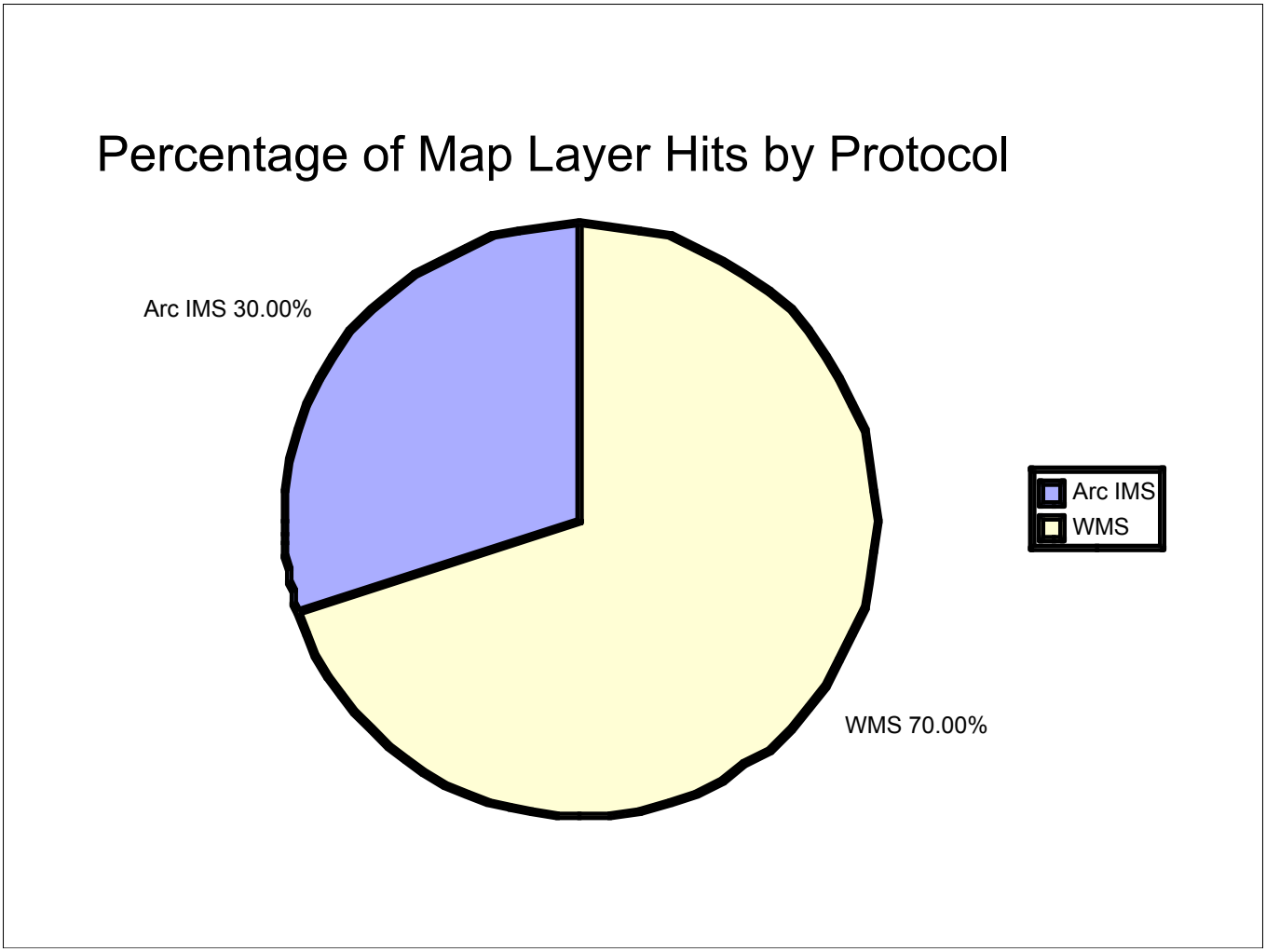
Financially, for the first deployment, the costs were similar, since Mapserver needed an initial capital investment in the FastCGI customization. (Licensing and maintenance costs for ArcIMS are approximations.) But as the number of servers scaled out, the financial drawback to ArcIMS gets much worse. In addition, since Mapserver is more efficient, it requires fewer servers for the same load. And because it's easier to administer, it requires less staff time.

What's "N"?

So, how big has the GeoBC WMS installation gotten?

openmaps.gov.bc.ca

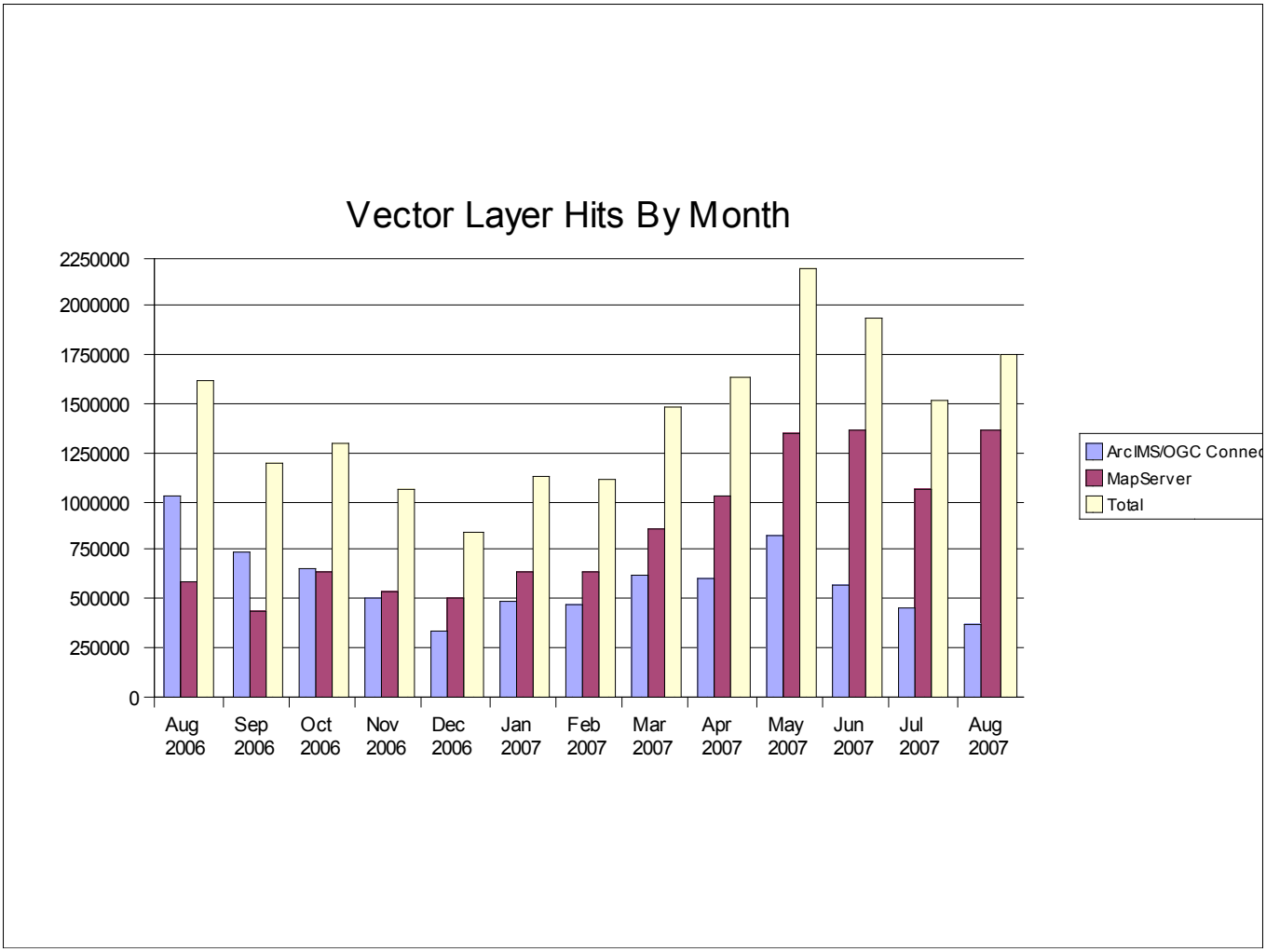
GeoBC provides most of their layers via WMS now.
At openmaps.gov.bc.ca, you can see all the services BC provides as WMS services.
Here's a topographic map,
And a cadastral map.



The WMS services were put into production in 2005. At that point, all the web mapping was ArcIMS-based for custom web apps.

At this point, two years later, the WMS services are now being used more than the old ArcIMS services by a factor of two.

Public access to standardized web services like WMS is more effective and popular than access to data via custom web interfaces (which is what the ArcIMS systems sit behind).



As time goes on, Mapserver is serving more and more of the GeoBC traffic, because the Mapserver installations are being expanded, while the ArcIMS ones are being kept in maintenance for old ArcIMS-only applications.

Operational wins

- Low cost to scale
- High standards support
- Easy administration
- Enhancements cost less than capital cost of ArcIMS
- No mandatory annual maintenance fees

Mapserver has had substantial operational wins.
- Low cost to scale
- Good and improving standards (WMS 1.3 coming in next release)
- Easy administration
- Low cost for enhancements (have since also improved the Mapserver ability to do SDE table joins, for under $10K).

Bonus win!

- In 2006, a bad ArcSDE patch took down the British Columbia data warehouse for a week.
- Because the WMS systems were based on Mapserver, the support team were able to migrate the data to a temporary PostGIS instance and serve it from there.

An added bonus, support for multiple data sources.
ArcIMS supports ArcSDE only as a database backend.
Mapserver supports PostGIS, Oracle Spatial, MySQL, and ArcSDE.

Support?

- Services needed
  - Windows compiles with ArcSDE
  - Occasional enhancements
- Time-and-materials contract
  - Refractions Research, local skills

In order to use Mapserver operationally, they needed some support.
They deploy on Windows servers, so need custom builds with ArcSDE and new patches.
They need occasional enhancements (like SDE joins).

GeoBC simply put a services contract in place, and pay for the hours they use.
Some years it's higher, some years it's zero.

Should we use open source?

- If it meets your needs!
- Test it!
- Make sure!
- Ask an expert for help in testing

So, if you run an operational organization, should you use open source?
Maybe!
But the GeoBC experience gives some lessons
- do empirical testing, because the theory of the product might not equal the practice
- have an expert help with testing, so that misconfiguration or mistakes don't bias the result
- consider improving the product if it comes up short in fixable ways, use your expert to define "fixable"

What to evaluate?

- Technical factors
  - Compatibility, speed, administration
- Non-technical factors
  - Community, responsiveness, commercial actors

- In addition to technical testing, evaluate the strength of the community behind the product
- Activity of mailing lists, existence of companies in the field
- Development and recent releases, is development ongoing?

This whole process was kicked off by the need/desire for WMS services in BC, which was in turn promoted by a national strategic body, GeoConnections.
The strategic can feed the operational!

GeoConnections is a development program of the federal government
o first, to collaborate with provinces and territories in making Canada's location-based data more accessible and compatible;
o second, to collaborate with the private sector in developing technologies to share this data over the Internet;
o third, to create the partnerships and conditions required to build a national infrastructure; and
o four, to work with the public sector in developing the policies required to share data.

- - $60M over 6 years 1991-2005
- - $60M over 5 years 2005-2010
- Cost sharing always 50/50 max

GeoConnections is primarily a funding pool, and it has been well funded.
The money in GeoConnections is not for open source, it is for geospatial development.
Sometimes it goes to open source, usually not.
GeoConnections dollars always have to be cost shared with other dollars.

- Operational initiatives
  - Funding services
  - CGDI
  - Provincial and others
  - National SDI frameworks (NFIS, WLIS)
- Strategic intiatives
  - Funding technology (GeoInnovations)
  - Funding data (GeoBase)
  - Choosing standards (Technical Steering)

None of GeoConnections mandates has to do with open source.
GeoConnections is trying to build a national SDI, from the ground up, by giving the local and provincial organizations the financial incentives to do the right things in terms of policy and technology.

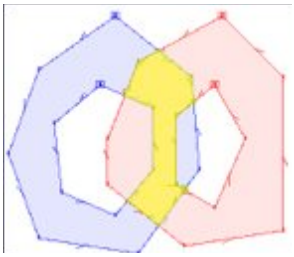
Strategic

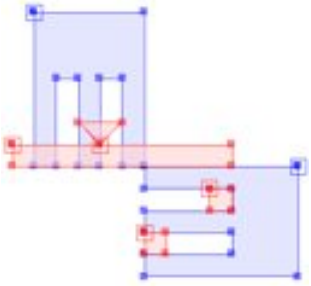
GeoBC also has an interesting strategic story, which has had huge international implications because of open source.

GeoBC (at the time, MSRM) put in a proposal to GeoConnections for cost sharing dollars.
GeoBC looked at future systems needs and recognized they were missing a fundamental building block.
Which would cost a lot to either build or license, if it did not exist.

Java Topology Suite

- British Columbia proposal to GeoConnections
- Java software implementation of OGC simple features
- Basic software building block





They proposed a “topology suite” a core Java library of geographic objects and methods for building applications.
They proposed Java for platform independence.
They proposed open source for flexibility of re-use.

Really stupid idea

It’s hard to overstate just how dumb the idea of building a foundational geometry library from scratch is, for a government agency.
Their job is collecting and publishing data, not writing software!
However, they pushed the idea, as much for the love of the idea, as for the strategic value.
They saw that without a solid foundation, nothing non-trivial could be built later on.

Java Topology Suite

- Cost ~ $250,000
- Delivered 1.0 with full OGC support
- Immediately ported to C++ by 3rd party for use in PostGIS

GeoConnections approved the project, and about $250K was spend on it.

The utility of the algorithms immediately apparent,
Refractions arranged for the algorithms to be ported to C++ for use in PostGIS.

JTS has also been ported to .Net for use as a foundational library in that language.

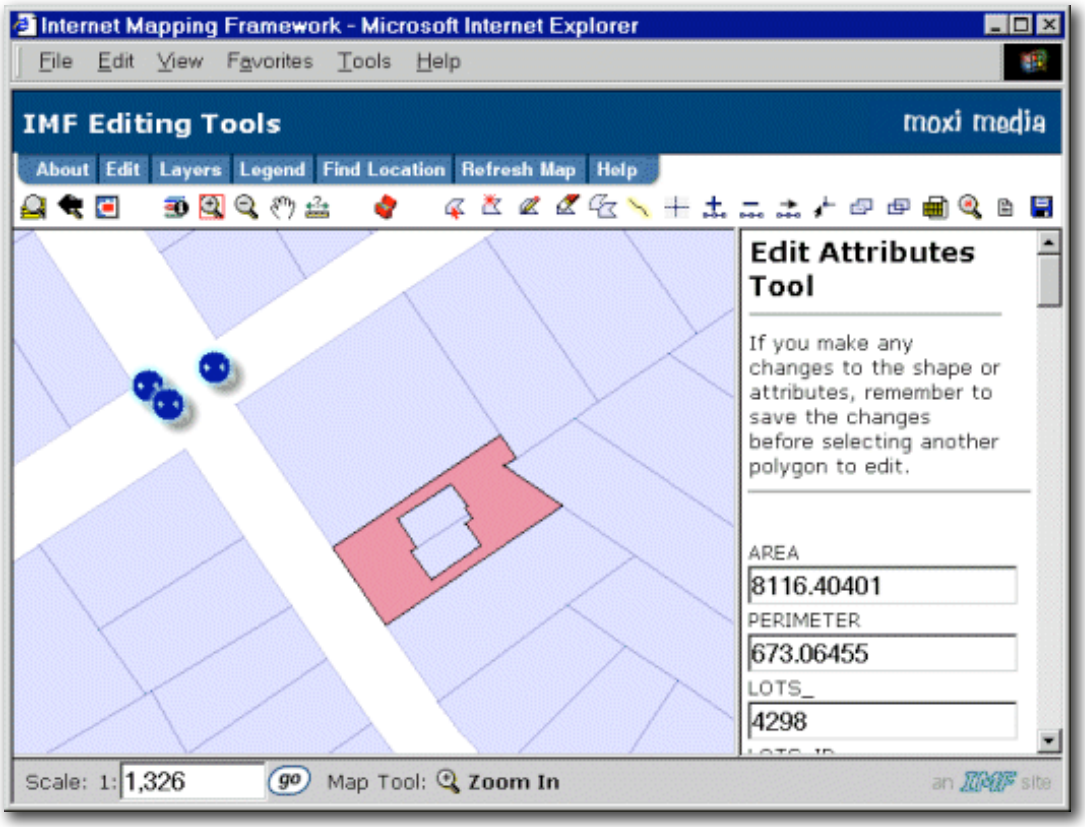
Internationally, JTS has been used in hundreds of projects, inside packages like PostGIS, GeoTools and Geoserver, and as a standalone library for data processing.

From a global perspective, the cost of developing JTS has been re-couped many many times over.

Stupid?

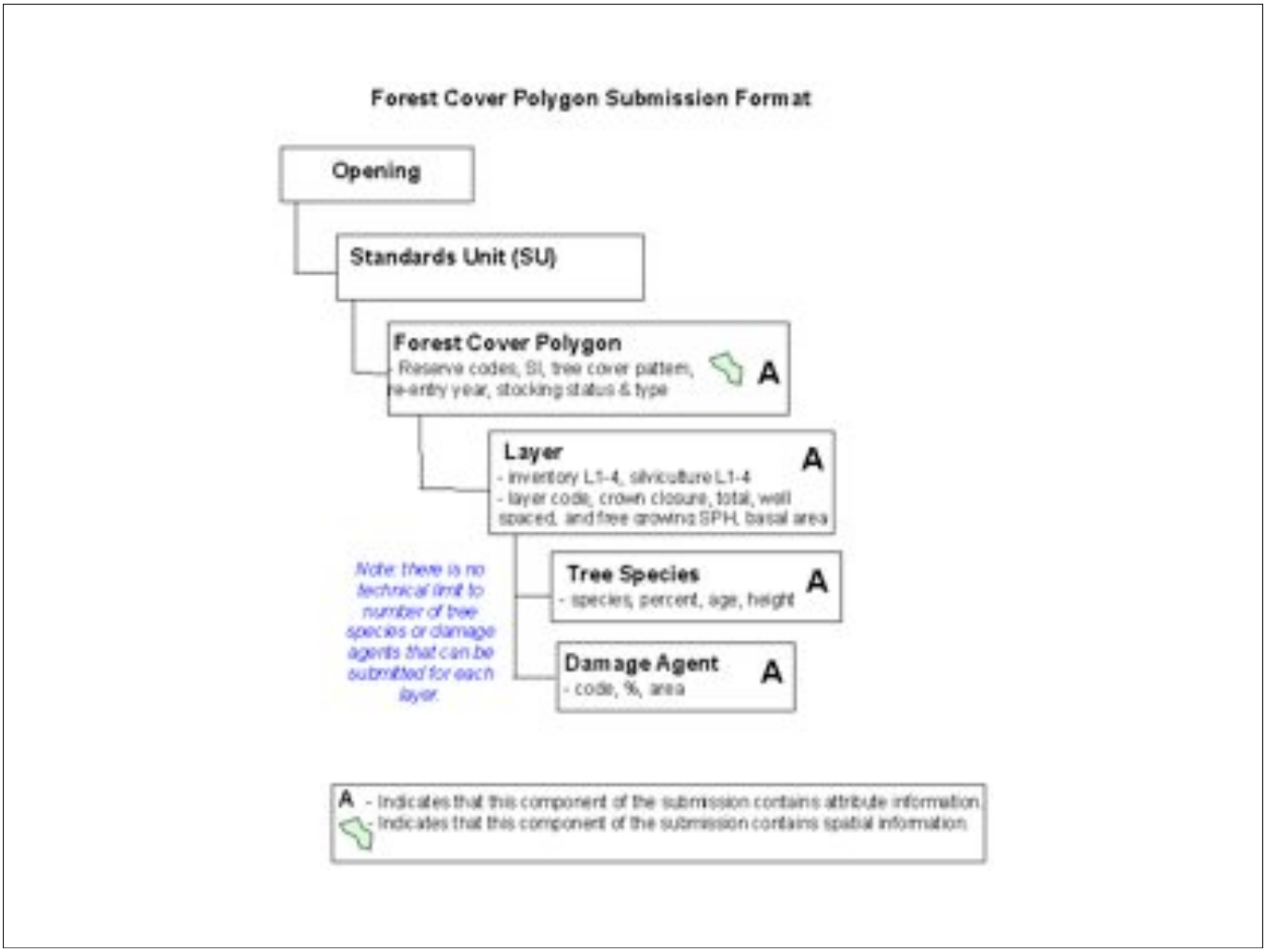
Penny wise == pound foolish

But what about from a BC perspective.
Would BC have been better to save their money?
Or would saving the development money have cost them more in the long run.
It has been enough years since JTS came out, that it is possible to take a stab at answering that question.

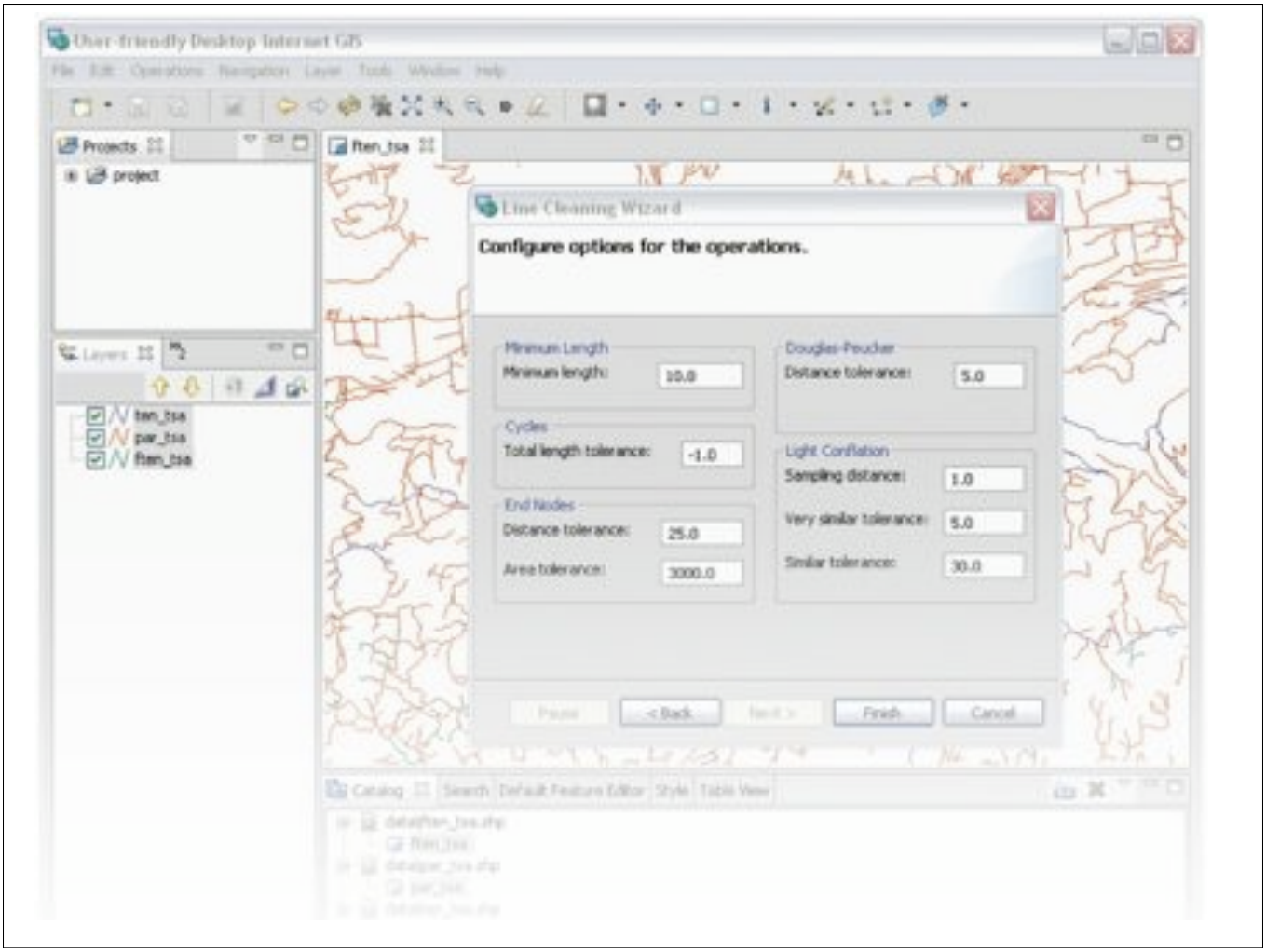


BC separately paid for the development of a web mapping framework, that runs on ArcIMS. ArcIMS doesn't provide all the hooks to spatial processing that the framework needed, so the developers used JTS to add the buffering and clipping functionality they needed.

Without JTS, they would have had to license the functionality or build it from scratch.

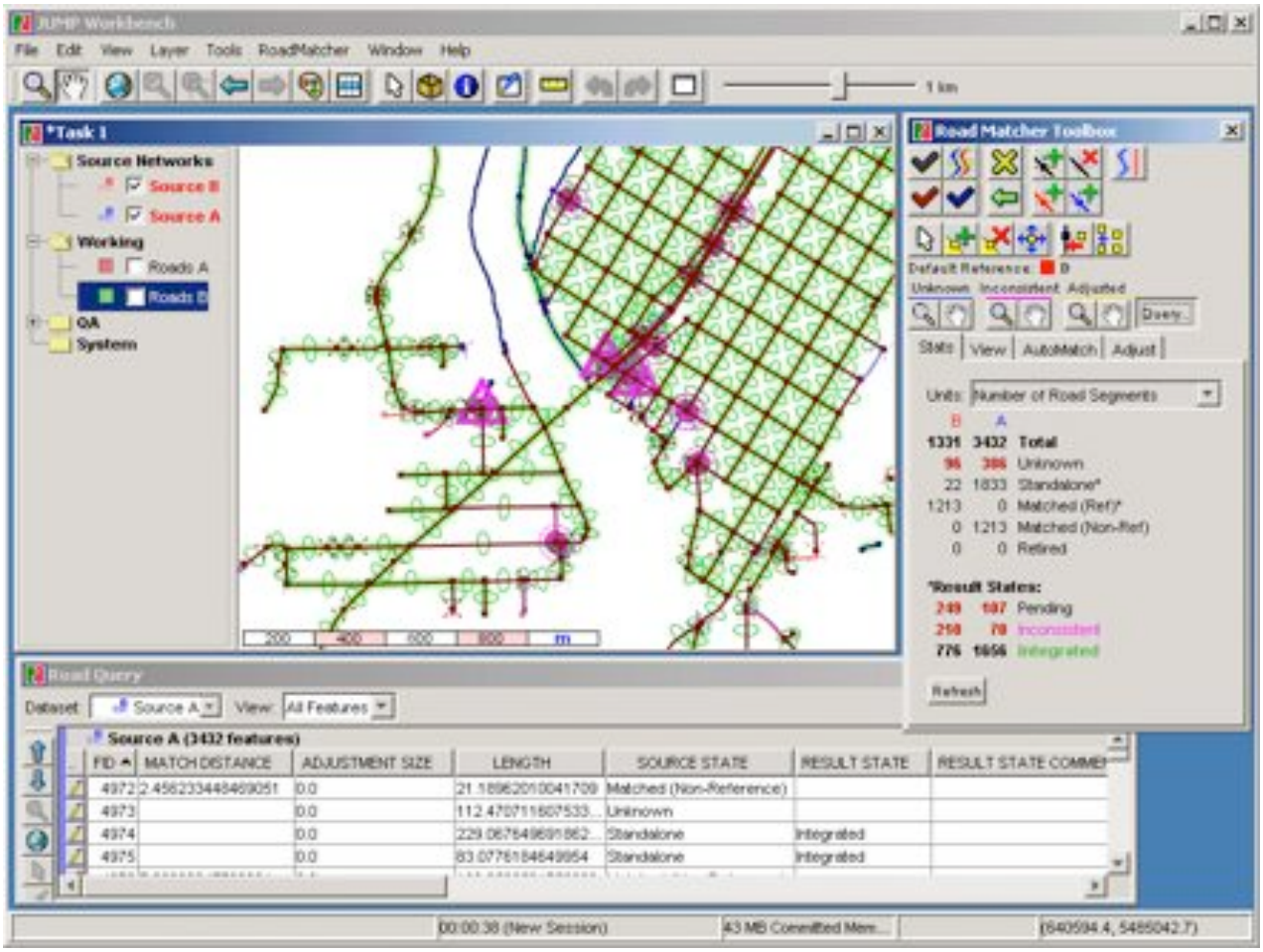


The BC Ministry of Forests developed a system for forest companies to submit cutting permit applications electronically, as a GML document.
The GML is validated spatially for topological correctness, using JTS.



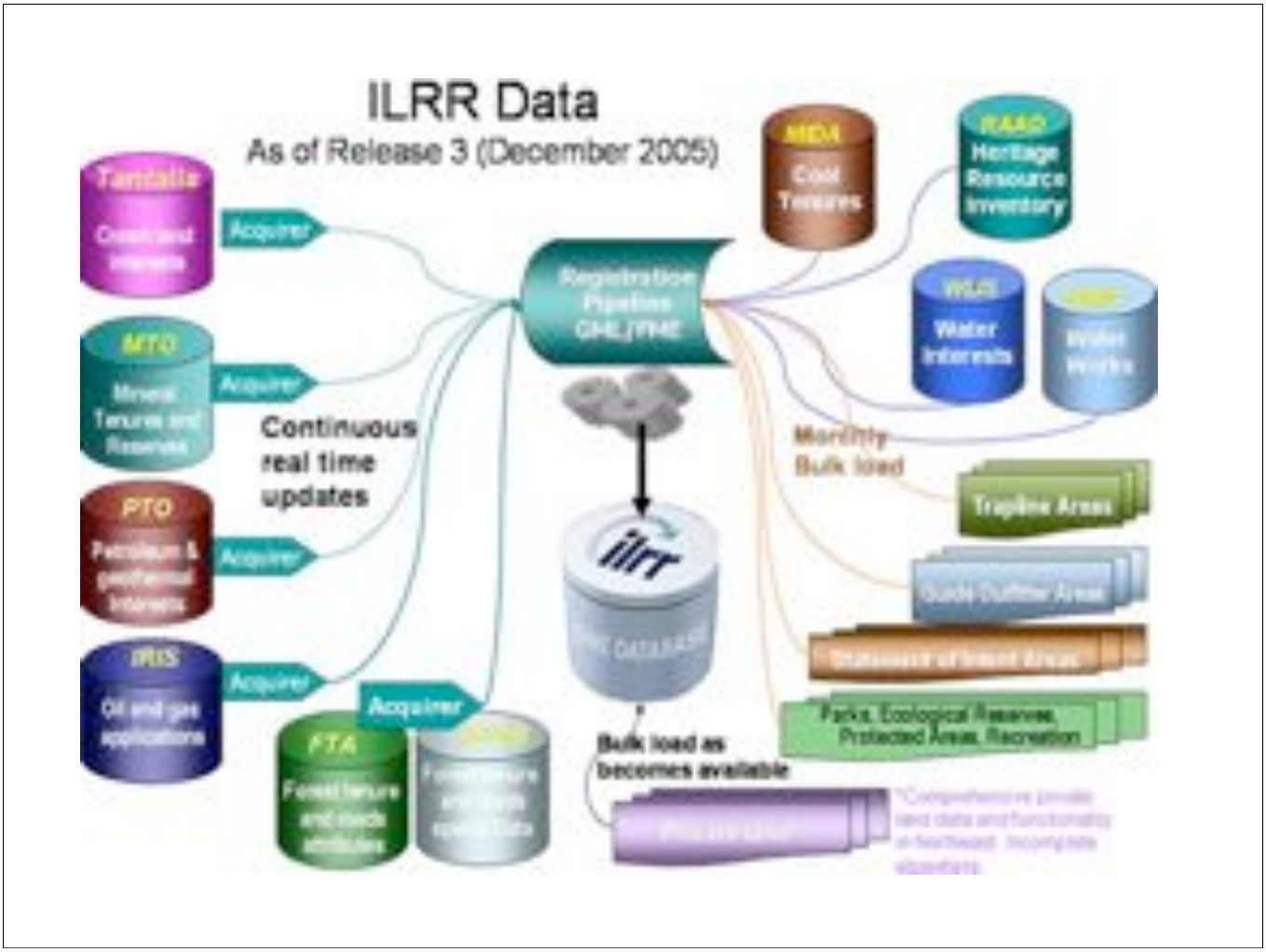
The BC Ministry of Forests also runs cut-and-yield models that require a clean connected road network.
They had a line cleaning routine built, that used JTS for line clipping and topological tests of the input data.

The Ministry of Forests Inventory maintains stand polygons for the whole working forest. As cut blocks are made, the blocks are cut into inventory coverage using the intersection algorithms from JTS.



The BC Digital Road Atlas has to integrate road network data from multiple sources (BC Elections, Ministry of Transport, Statistics Canada and others). The data conflation tool used to integrate the data is built on top of the JTS algorithms.

GeoBC has deployed Geoserver as a Web Feature Server (WFS) serving information about the digital photo library. Geoserver is built on top of the JTS geometry library and algorithms.



The Integrated Land and Resource Registry uses JTS in its source code as part of the GeoTools library.

Really stupid idea?

Even within British Columbia alone, the amount of development savings from the existence of JTS has been substantial. In addition, JTS has seen substantial use in Canada for other projects. The investment by GeoConnections has more than paid itself off.

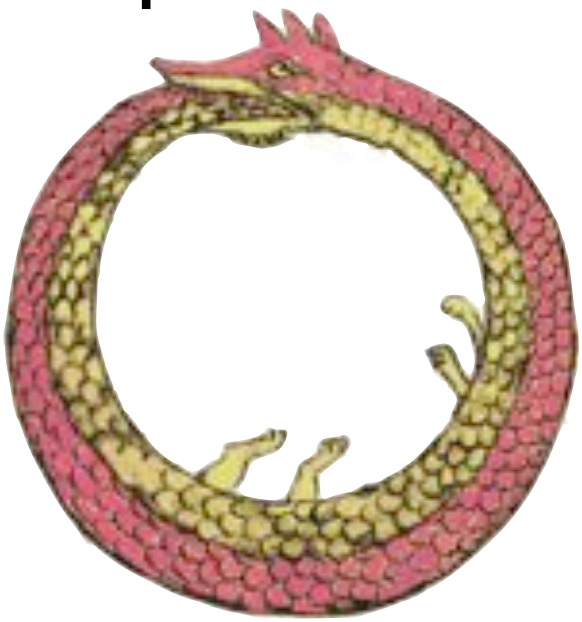
Third Party Improvements

As an open source project, JTS has also benefitted from outside contributions, which means that the Britich Columbia and Canadian projects using JTS have also benefitted.

- PostGIS community exercising algorithms
  - Test cases
  - Bugs found and fixed
- Direct funding for new development
  - Improved union

The port of JTS to C++ has allowed PostGIS to exercise the JTS algorithms. Failure cases discovered by PostGIS users have been fed back to JTS development and resulted in improved algorithms in JTS.
Funding for improvements to JTS has come in through the PostGIS connection, as PostGIS users look for improvements that are first implemented in JTS, then ported into PostGIS. The geometric union code received over $30,000 in funding from the Bavarian Ministry of Agriculture as they prepared PostGIS to receive their data sets and meet their processing requirements.

Operational



Strategic

At the end of the day, the strategic decision to fund JTS has come full circle, with JTS being used in a number of operational systems.
Some of the systems, like ILRR, aren't even aware that they are using software originally funded by the BC government for strategic purposes.