



Moving to the Cloud Isn't Difficult...

...Adapting to New Methods and Technologies is Hard

Ground Systems Architecture Workshop 2017
Los Angeles, CA

© 2017 by DigitalGlobe

Published by The Aerospace Corporation with permission.



This presentation focuses on advantages, pitfalls, and surprising discoveries along the way to a service enabled and cloud migration.

For more depth on the *architecture* we chose, please see our other presentation at this conference: [On The Road To Ground Systems That Scale For The Future](#)

Jon Upham, Director of Architecture and Innovation

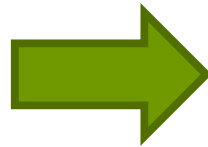
Mike Wierzbinski, Enterprise Cloud Architect

Mike Waters, Enterprise Software Architect

Email: GSAW-Correspondence@DigitalGlobe.com

Rearchitecture for Agility

DigitalGlobe recently rearchitected a large monolithic system, constructed over the last ~15 years, into a collection of microservices on a PaaS.



The new architecture:

- Flexible and adaptable
- New capabilities and speeds integration
- Allows Amazon Web Services

Brief Background on DigitalGlobe

- Started in Oakland, CA in 1992; currently Headquartered in Westminster, CO
- Owns and operates a constellation of satellites that image the earth: imagery collected $> 3,500,000 \text{ km}^2/\text{day}$
 - Slightly more than the landmass of India
- Downlink $\sim 5\text{-}6 \text{ TB}$ of new “raw” imagery per day ($\sim 2 \text{ PB}$ year)
- Create products from that imagery with various forms of image processing and analytics: $40 \text{ TB} - 100 \text{ TB}$ per day
- Customers include: Google, Microsoft, Apple, US Government, Friendly Foreign Governments

DigitalGlobe sets the standard for quality



Accuracy. Currency. Completeness. Consistency. We lead the industry.

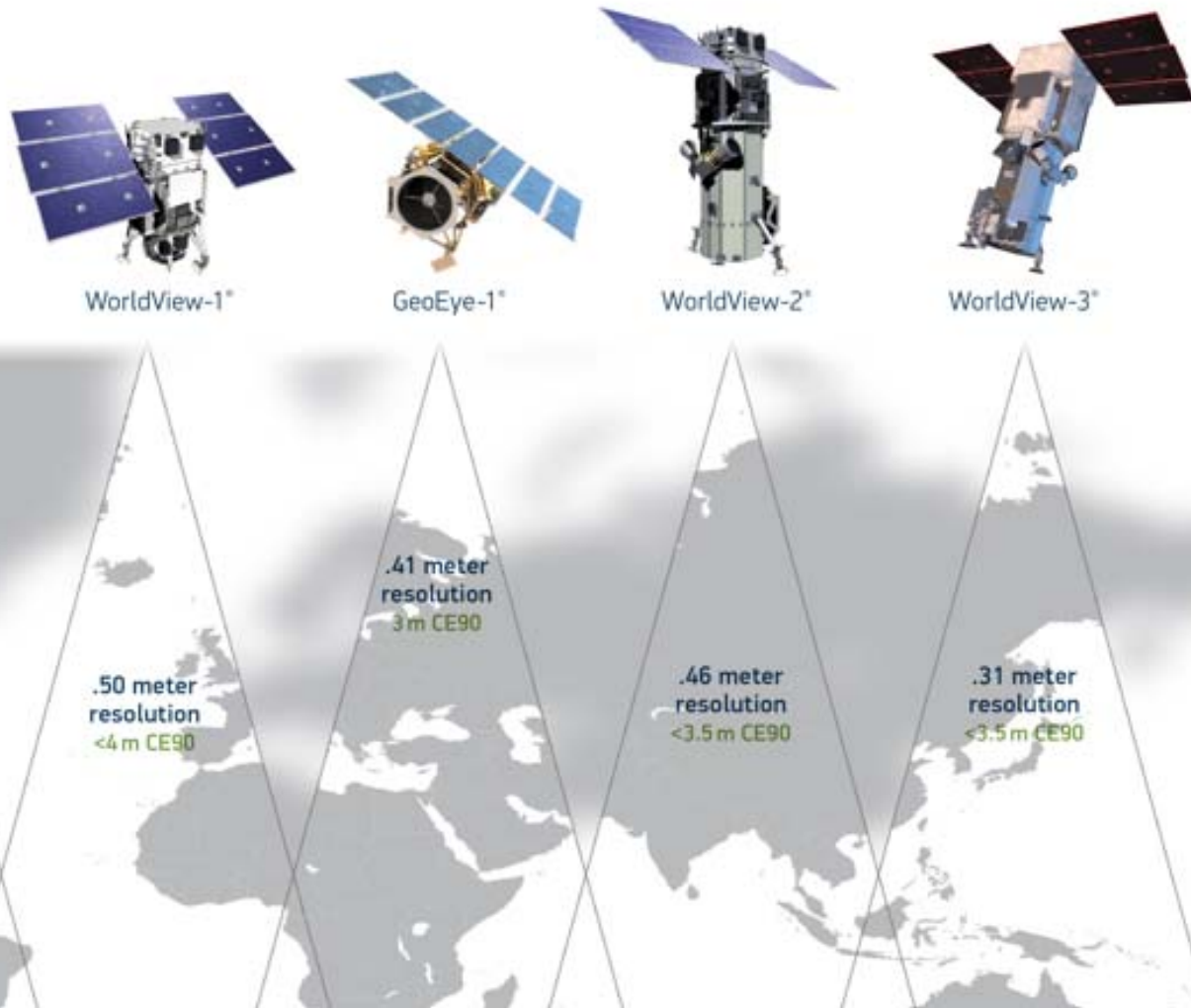


Orbit: 617 km
Resolution: 31 cm
Daily capacity: >400,000 km²
Agile: 10.6 s to slew 200 km

Launched: Nov 11th, 2016
First customer: Feb 1st, 2017

WorldView-4 just launched in Nov, 2016

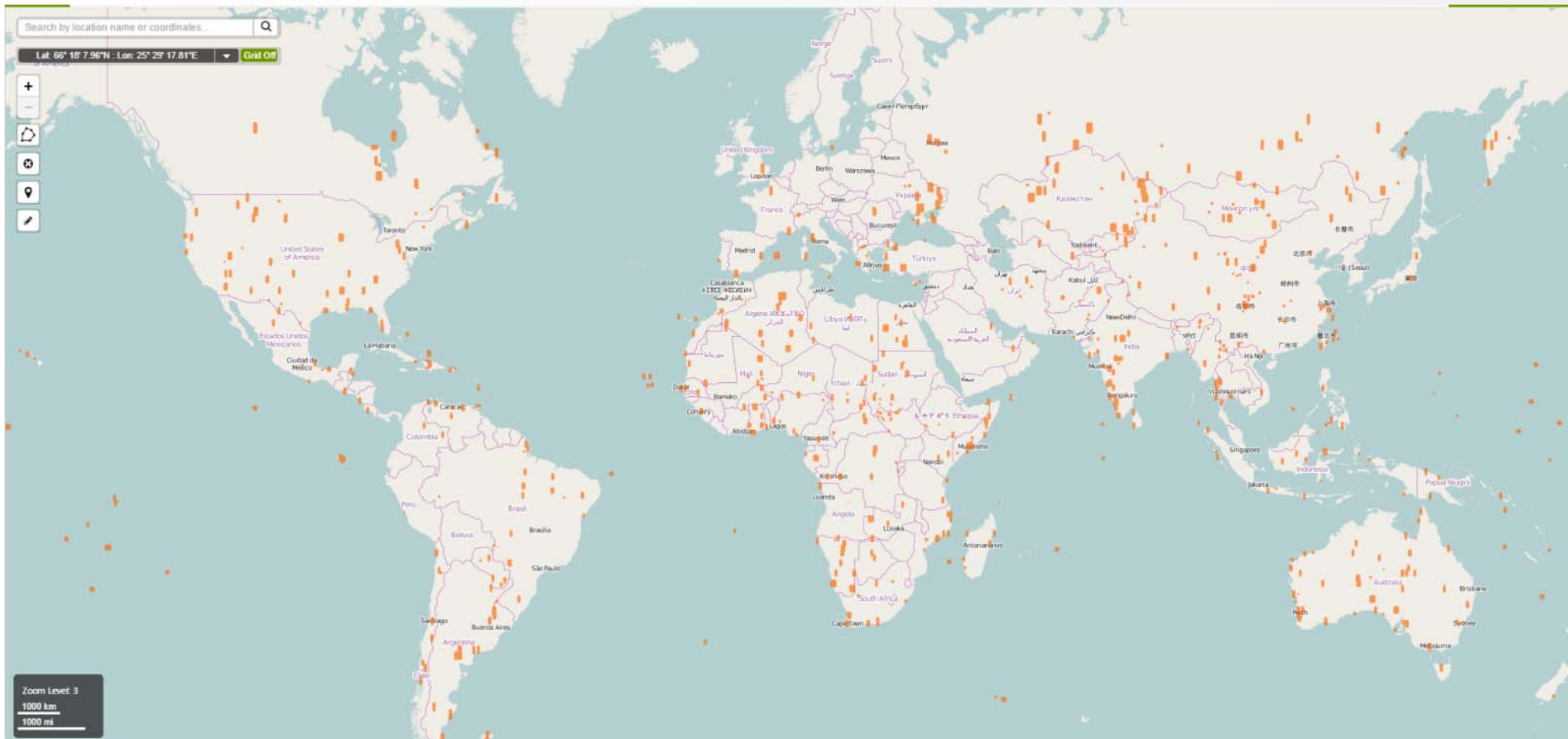
Accuracy. Currency. Completeness. Consistency. We lead the industry.



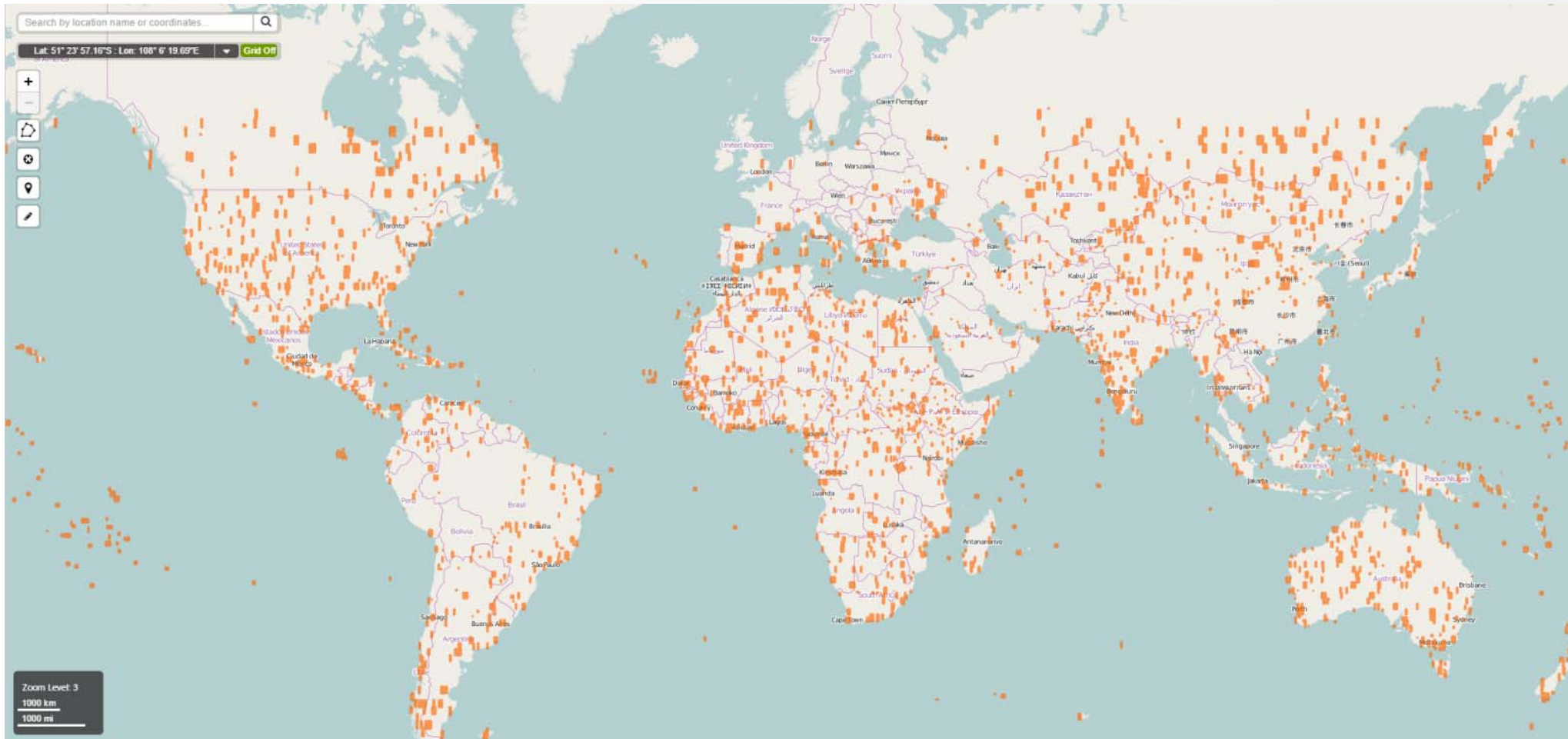
Orbit: 617 km
Resolution: 31 cm
Daily capacity: >400,000 km²
Agile: 10.6 s to slew 200 km

Launched: Nov 11th, 2016
First customer: Feb 1st, 2017

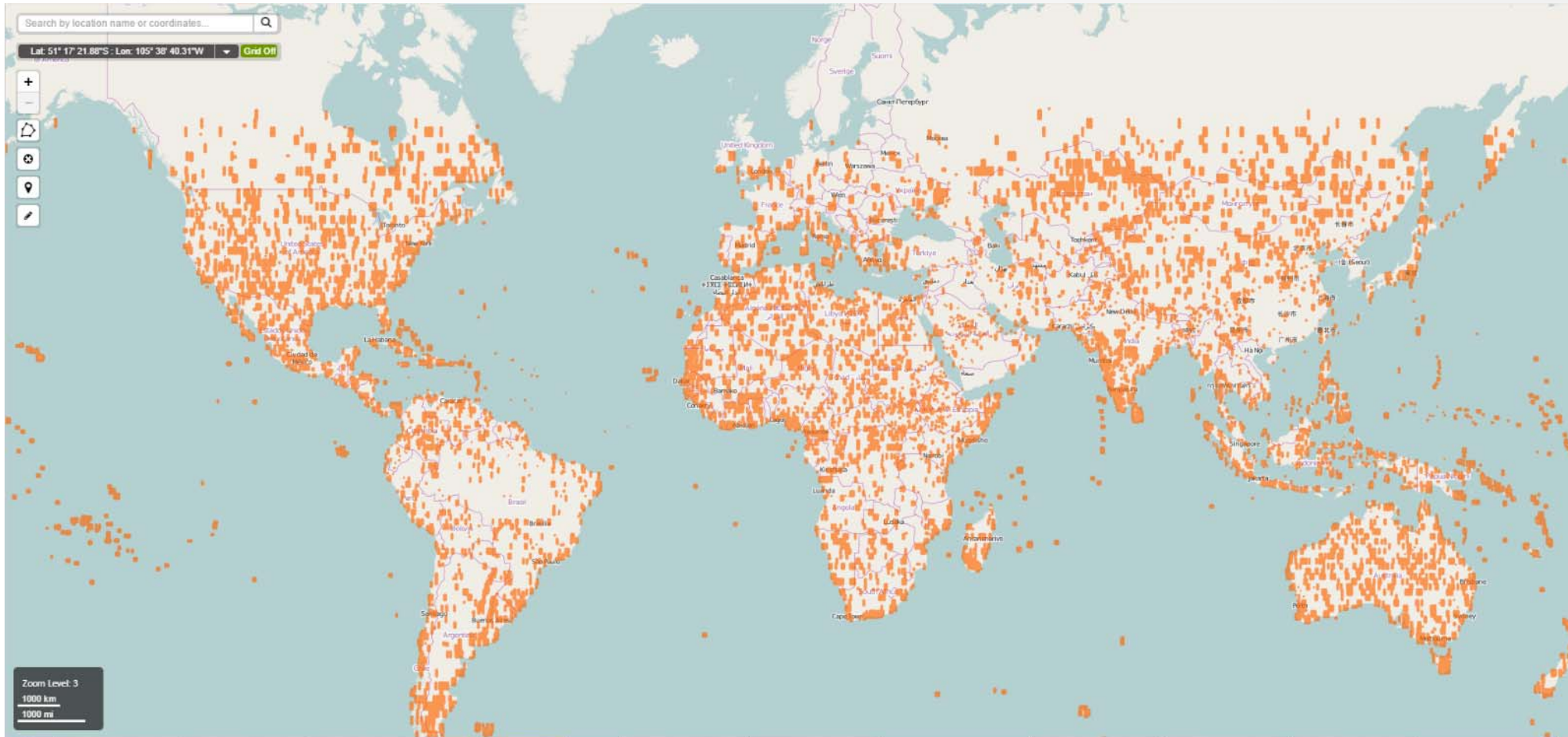
24 Hours of Collections



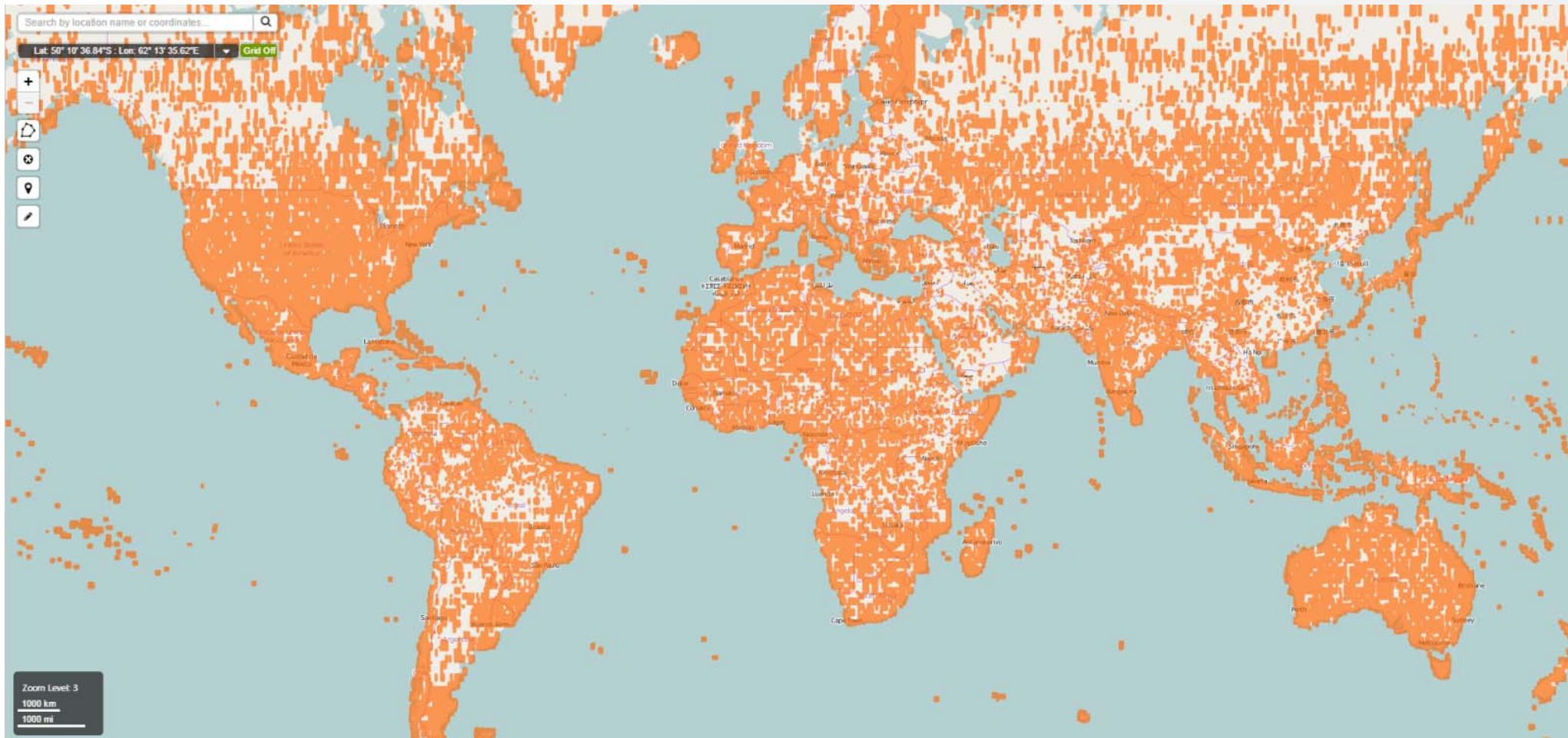
7 Days of Collections



1 Month of Collections



6 Months of Collections

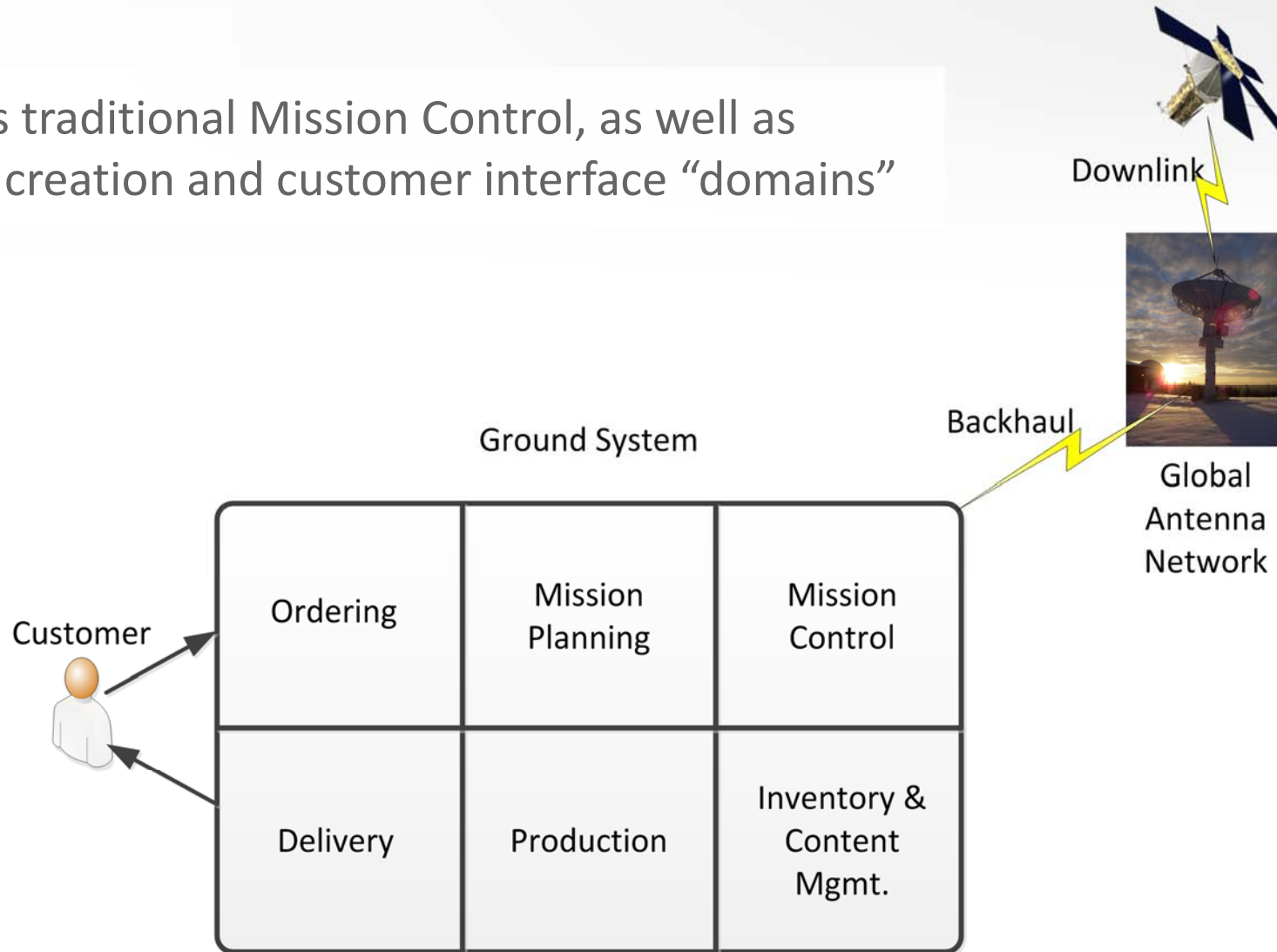


1 Year of Collections



The DG Ground System includes business functions

Contains traditional Mission Control, as well as product creation and customer interface “domains”



DigitalGlobe Needed A New Architecture



- Old Method –too complex
 - Dedicated hardware, long VM provisioning times
 - Large custom applications with central database and enterprise messaging
- New Opportunity – WorldView-4
 - “Go” in Q4 2014; Satellite launch scheduled Q4 2016
 - Designed and put in a microservice-based architecture with flexible/extensible workflows

The new system promotes agility

Services
created from
templates

HTTP/
RESTful
endpoints

Event-driven

Standards
Based

Services
orchestrated and
reused

Scalability and HA
via PaaS and IaaS

Continuous
Deployment

New capabilities don't have system-wide integration challenges

The rest of this presentation is a retrospective...

- We've just:
 - Attempted a huge transformation – and it's working!
 - Not complete, but well on its way, and pretty much everyone has bought into it at this point
 - Reached a watershed event – achieved operations and production
- So – it's a reasonable time to reflect on what we:
 - Have learned
 - Would have done differently
 - Want to do differently as we proceed

✓ Have a Strategy

✓ Build Support

✓ Get in Front Technically

✓ Watch out for Gotchas

✓ Have a Strategy:

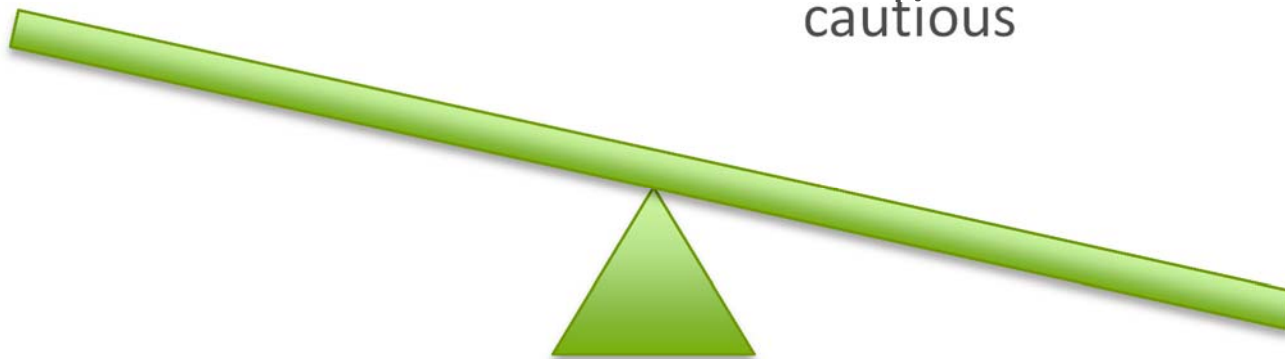
It will see you through hard times

- “Proof of Concept” / “Gradual Build”

- Shows capability
- Builds support
- Doesn't (usually) reveal issues
- Often throw-away
- Requires lots of expenditure in on-going integration
- “Double” support costs

- “All-In”: Business value can stand alone in new system

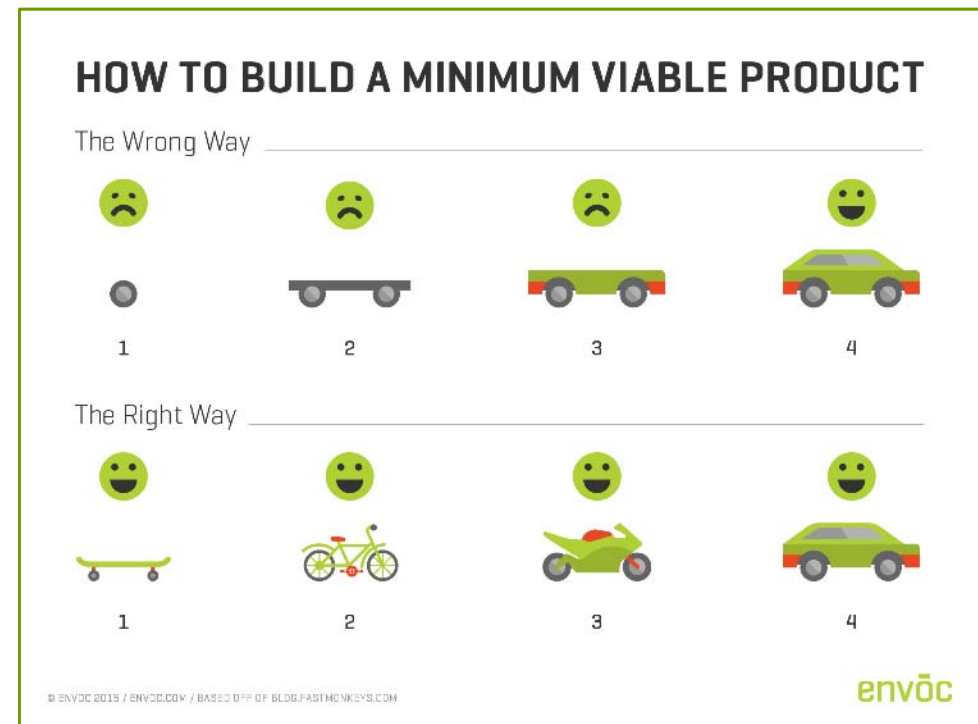
- Invest in Future
- Reduce bridge/scaffold to legacy
- Reduce investment in legacy
- Quicker transition / Lower Operating Expense
- ✦ High Risk / High Reward
- “Open Season” on the decision for those that want to be more cautious



✓ Have a Strategy:

Use “MVP” concept at Macro Scale

- Delivering demonstrable value with each iteration is the goal
 - Big-bang (“You don’t get anything ‘til you get everything”) components caused some significant churn
 - Again: Demo/Demo/Demo
-
- Break project into useful, stand alone iterations that each bring business value
 - Ask yourself, if the project were cancelled at anytime along the way, would I have anything worthwhile?



✓ Build Support: Tendency to turn back

- You can bet that no matter how bad your current architecture is, there will come a time during the project where there will be a wave of people that yearn to return to the system that they know
- We reminded ourselves of Cortez / Alexander the Great
 - “Burn the Boats”
- Seek support at three levels:
 - Executive
 - Benevolent Dictator
 - Lead Developer



✓ Build Support: Top cover

- Service and Cloud enabling an existing business is bold, and will have detractors
- Representation on executive row is important!
- When “shopping for an executive”, focus on the following qualities:
 - Understands both the political and technical *challenges*
 - Understands the *advantages* and can articulate them well
 - Business-oriented / Results-oriented
 - Able to stand strong when everyone wants to go back to the way it was



✓ Build Support: “Ruthless Ruler”

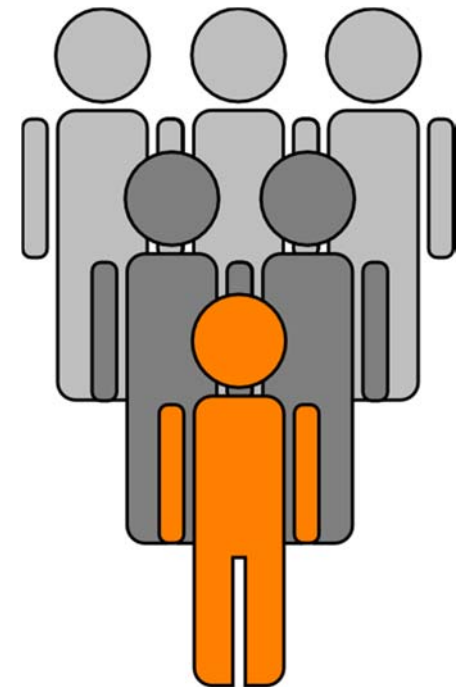
Final Technical Authority: we didn’t have this – we wish that we had

- Reduce/eliminate the “less filling / tastes great” debates when considering alternatives (more on this later)
- Take the heat and the responsibility, reducing the angst within the development group(s)
- Review and enforce technology decisions
- Encourage (demand?) standardization



✓ Build Support: Recruit Lead Developers

- Programmers are quite independent – but they
 - Respect technical expertise
 - Tend to pay attention to solutions recommended by people that they respect
- Gaining buy-in from lead developers means
 - You have someone ‘on the inside’ ready to defend the new direction
 - Head-off (or at least identify) misunderstandings before they get out of hand



✓ Get in Front Technically:

- There will be lots of questions about “*how*”; it’s the leader’s job to be several steps ahead
- Think about:
 - Architectural Runway and Macro-Level System Design
 - Prototyping Iteratively
 - Including Customers and Users Early

✓ Get in Front Technically:

Runway & Macro Design

Architectural Runway: Do your homework

- Architecture spends time to evaluate / Quick & Dirty prototype / reduce the field of design alternatives
- Cut down on the number of options that must be discussed with the technical team

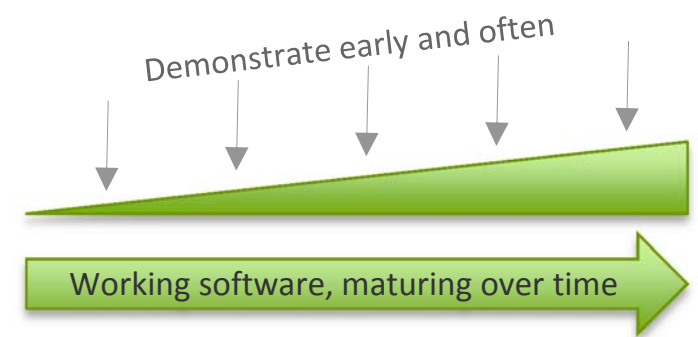
Macro Level System Design

- Focus on a particular domain, service, or component
- Get the best minds in a room to hash it out: limit the audience to 'players' (not spectators)
- Resulting design documented as basic interface definitions and high level sequence diagrams—avoid getting sucked into the details
- Focus on progress and velocity – Goal was: 80% right / 20% rework



✓ Get in Front Technically: Prove it early

- Pioneer team – “road-tested” the prototypes from Architecture
- Prove to ourselves that the teams could ‘get it’
- Progressively more complex End-to-End demonstrations on short timelines
- Gain feedback, and show notable progress
 - Start with the “Hello World” End-to-End demonstration
 - Evolve it and add on to it until you have a system that is Launch-ready



✓ Get in Front Technically: Engage Users

Users and Operations

- DevOps – we still don't have it nailed
- Lots of opinions about what DevOps means – made more difficult by 24x7x365 character of our business
- End-users initially unavailable (other commitments) – We engaged with them early, but didn't do enough to pull them in and keep them engaged throughout
- UI's are a sore point that we are addressing now



✓ Watch out for Gotchas:

Complexity & Communication

Complexity of something new

- New systems bring new processes—don't underestimate the details involved in: CI/CD, vulnerability scanning, code-repos, test automation, and learning to code in micro-services

Communication Everywhere

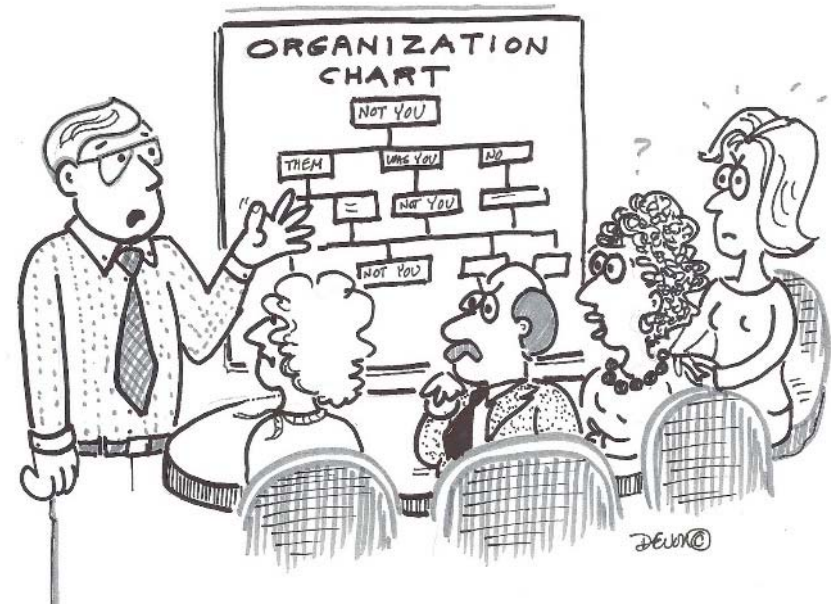
- Cross-enterprise vision casting (with audible support from the top)—Energy spent here pays dividends!
- Consistent and continuous communication is key: Developer's Round Table
 - We used a cross-enterprise Developers' Roundtable forum to ID problems and thorns, seek input, make decisions and move on



✓ Watch out for Gotchas: People!

Organizational Issues

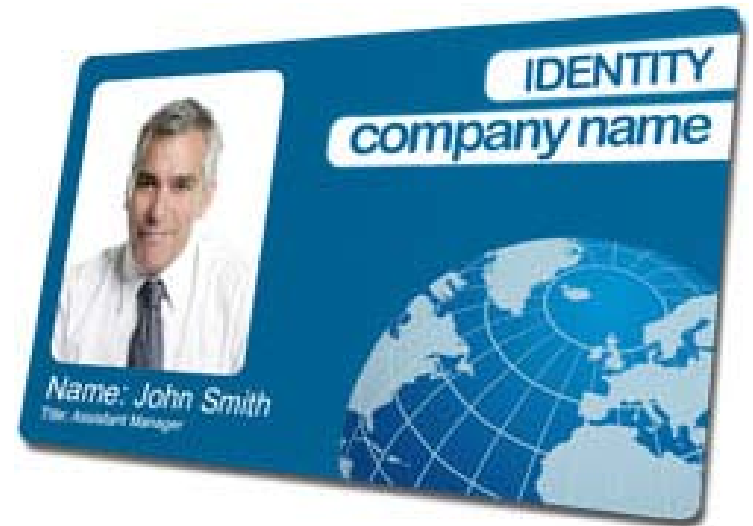
- Discontinuity of roles can tax organizational velocity
- Decentralize where you can, and balance listening to team concerns with taking responsibility off their shoulders
- “Politically” motivated technical decisions do happen
 - Enlist analysts and others that have been there
 - Prototype to prove capability
- Staff Up where headcount will be saved: cheaper long-term is usually more expensive short-term; be prepared to expend energy to educate



The final ingredient: Passion

If a re-engineering project of substantial breadth and impact is going to succeed...

Someone needs to be passionate enough about the solution to put their badge on the line



There is no 'good time' for re-architecture

- We have gained:
 - Velocity
 - Flexibility
 - Faster cycle time
 - Business Opportunities
 - More reliable deployments
 - Reduction in operating expense
- This will look good in the rear-view mirror – but while you're in it:
 - Volatile
 - Divisive
 - Challenging



- But so worth it!

Purpose

Seeing a better world.

By giving our customers the power to see the
Earth clearly and in new ways, we enable them to
make our world a better place

For more depth on the *architecture*
we chose, see our other presentation:
On The Road To Ground Systems That
Scale For The Future

Email: GSAW-Correspondence@DigitalGlobe.com