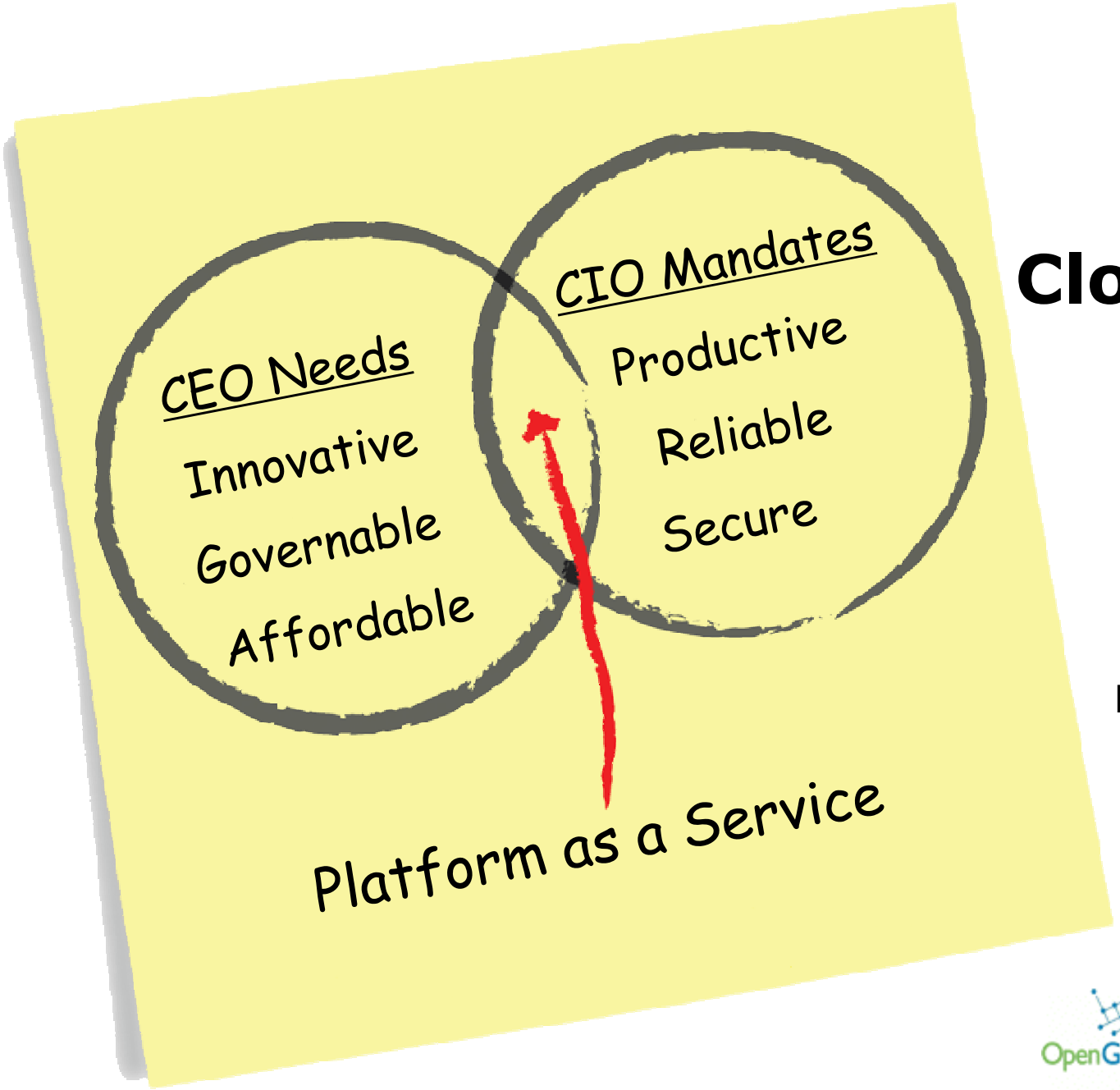


Cloud Computing:

More Than a Virtual Stack

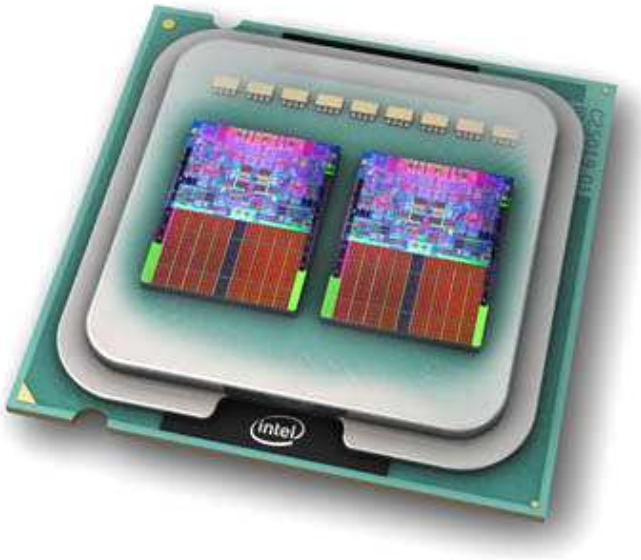
Peter Coffee
Director, Platform Research
salesforce.com

Torrance, California, USA
pcoffee@salesforce.com

It’s Getting Awfully Cloudy Out There

What Makes the Cloud Compelling

- Since the IBM PC was introduced
  - Processor speed has risen 30 per cent per year
  - Memory capacity grown by 50 per cent per year
  - Mass storage mushroomed 80 per cent per year
- Desktop systems are burdened with too much state
  - File system technology has not addressed new needs
  - Governance of critical data falls short of rising demands



- Trends redefine “best practice”
  - Bandwidth has grown 40 per cent per year
  - Processor performance trends favor shared machines
  - Data centralization improves coherence and governance

What Makes the Cloud Compelling

- Since the IBM PC was introduced
  - Processor speed has risen 30 per cent per year
  - Memory capacity grown by 50 per cent per year
  - Mass storage mushroomed 80 per cent per year
- Desktop systems are burdened with too much state
  - File system technology has not addressed new needs
  - Governance of critical data falls short of internal demands
- Trends redefine "best practice"
  - Bandwidth has grown 40 per cent per year
  - Processor performance trends favor shared machines
  - Data centralization improves coherence and governance

"We expect to see, by 2012, [20 to 25 per cent] of the server market will be running some version of cloud computing…
"Right now, as much as 14 percent of server purchases are going into some sort of cloud deployment."

Jason Waxman
General Manager, High-Density Computing
Intel Server Platforms Group
17 Feb. 2009

To Qualify as a Cloud

- **C**ommon, **L**ocation-independent, **O**nline **U**tility on **D**emand*
  - **Common** implies multi-tenancy, not single or isolated tenancy
  - **Utility** implies pay-for-use pricing
  - **on Demand** implies ~infinite, ~immediate, ~invisible scalability

- Alternatively, a "Zero-One-Infinity" definition:**

  0 On-premise infrastructure
  Acquisition cost
  Adoption cost
  Support cost

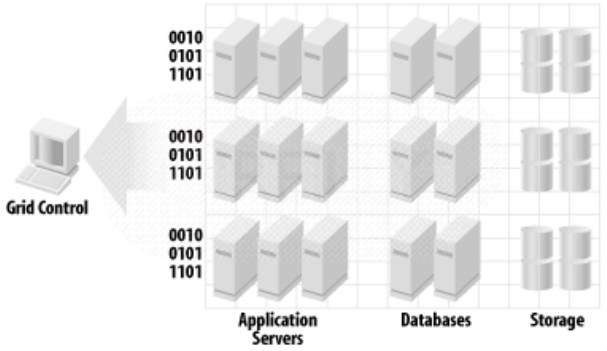
  1 Coherent and resilient environment – not a brittle "software stack"

  ∞ Scalability in response to changing need
  Integratability/Interoperability with legacy assets and other services
  Customizability/Programmability from data, through logic, up into the user interface without compromising robust multi-tenancy

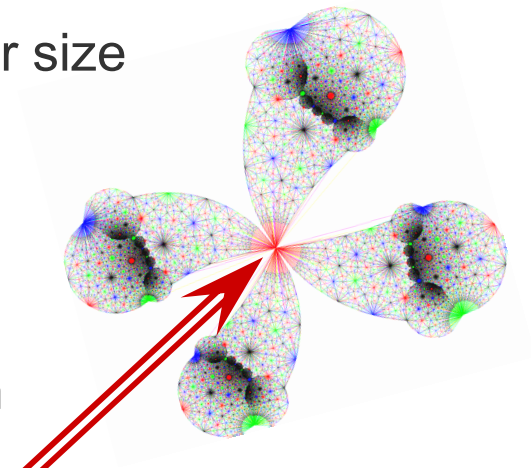
* Joe Weinman, Vice President of Solutions Sales, AT&T, 3 Nov. 2008

** From The Jargon File: "Allow none of foo, one of foo, or any number of foo"

Clouds Aren't All the Same



- Not every cloud is a "grid"
  - Grids imply dynamic arrival/departure
  - Electrical analogy has limits: CPU cycles aren't substitutable
- Most clouds are not "compute clusters"
  - Clusters are typically monocultures: just one type of node
  - Applications may require tuning to a particular cluster size
- Some clouds are servers in virtual slices
  - Virtualized servers *can* be quickly provisioned
  - Spin-up of instances = new management task
  - Hardware gets cheaper, management…not so much



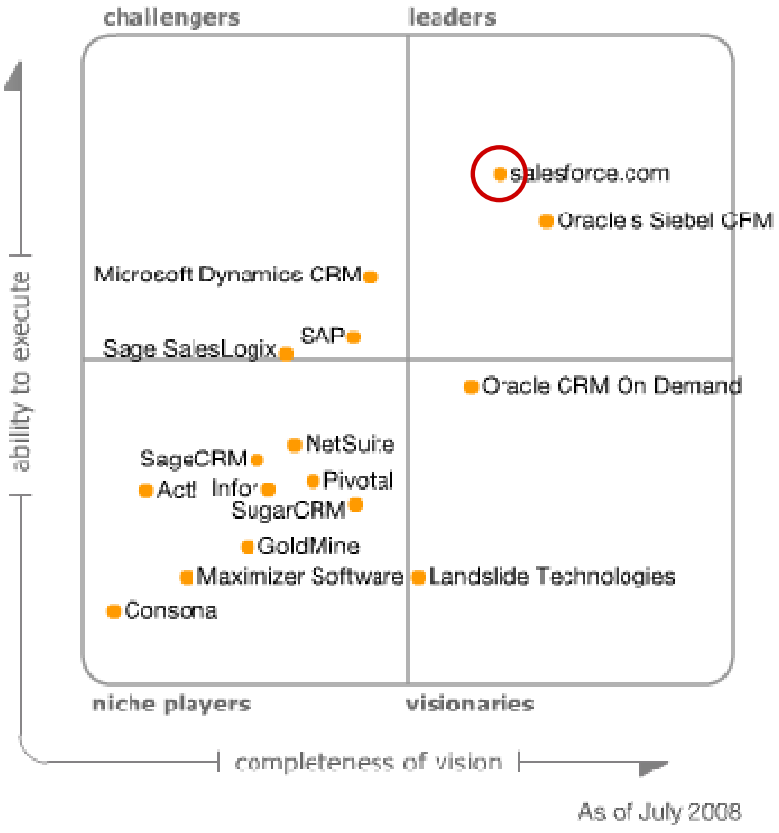
- **Enterprise cloud computing implies API leverage**
  - Immediate focus on **function**; immediate delivery of **value**
  - Using appropriate frameworks enables a **huge head start**

Our Cloud Began with CRM

- **Fundamental ideas**
  - Enterprise software should be as accessible as the Web
  - Web-based systems should be designed for global scale
  - Everything that's not distinctive to a customer should be shared
  - Everything that's distinctive to a customer should be customizable
- **Logical implications**
  - Multi-tenant architecture
  - Metadata-based customization
  - Transparent upgrades
  - Ease of adoption enables focus on continued improvement
- **Results**
  - **Mainstream assimilation**
  - **Customer success:**

    92% "would recommend"

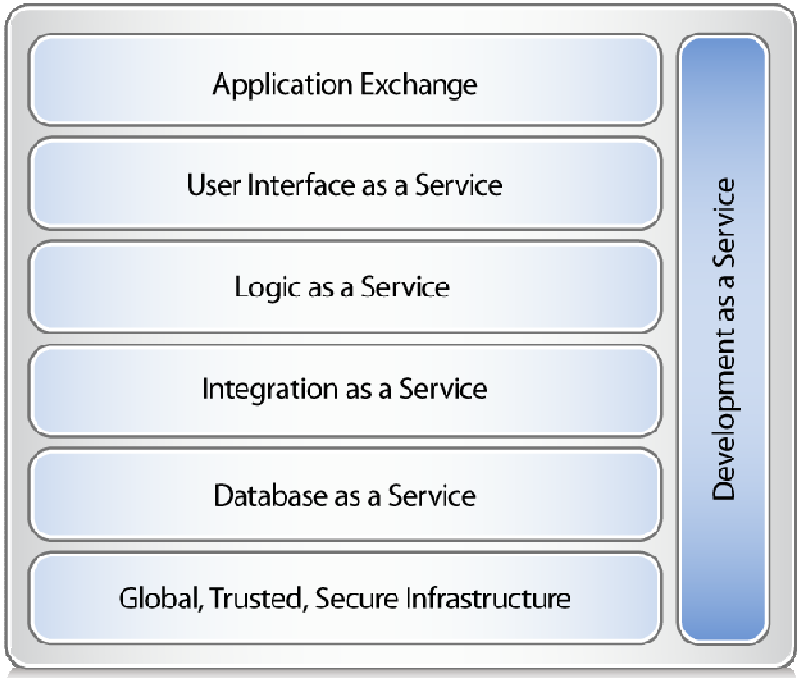
    77% have already done so



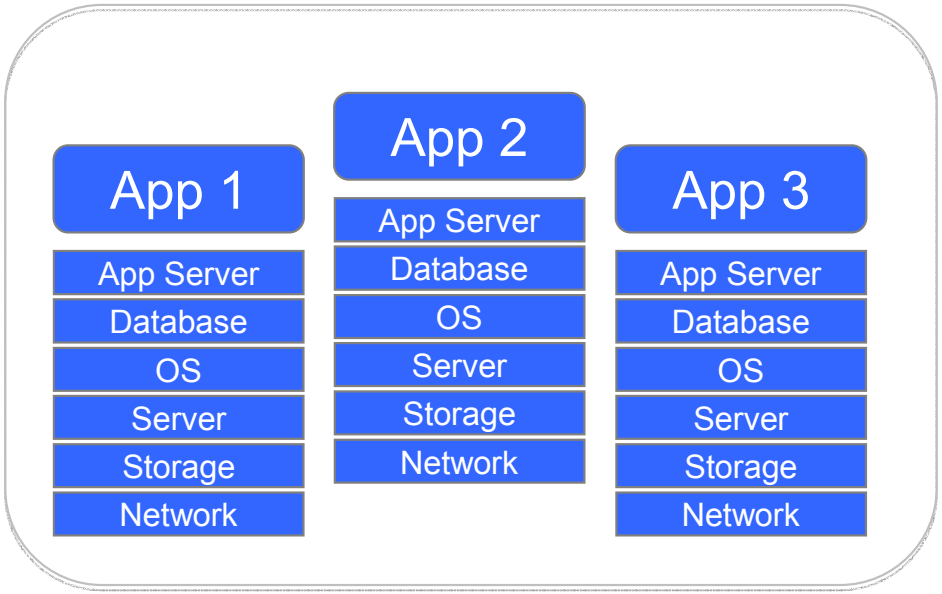
A Customer-Driven Platform

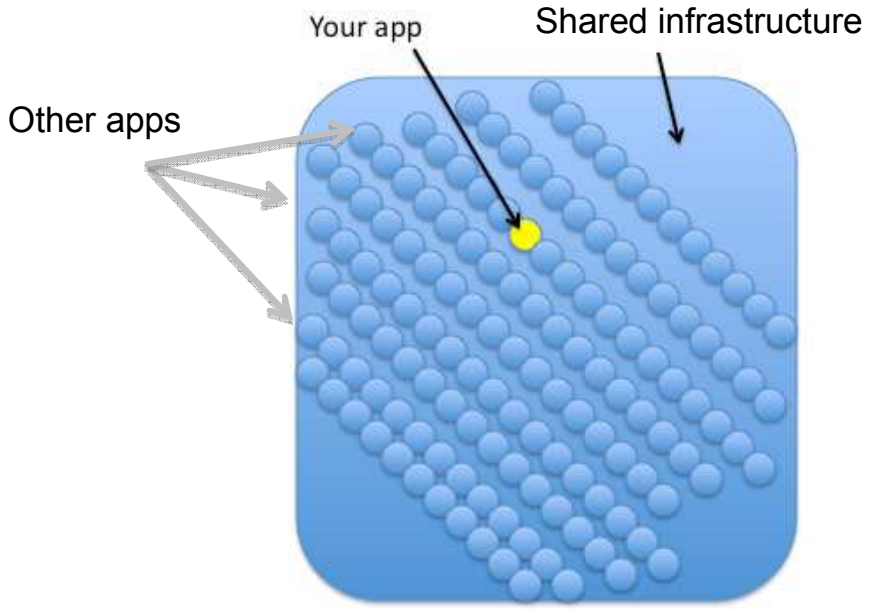
- **Customers wanted more**
  - More customization
  - More integration
  - More power to automate and extend

- **Clean-sheet architecture sped change**
  - 28 releases in ten years
  - All customers on current version
  - Web standards-based ecosystem
- **Results**
  - **Platform capability**
  - **New options for enterprise IT**



Single-Tenant vs. Multi-Tenant Architecture

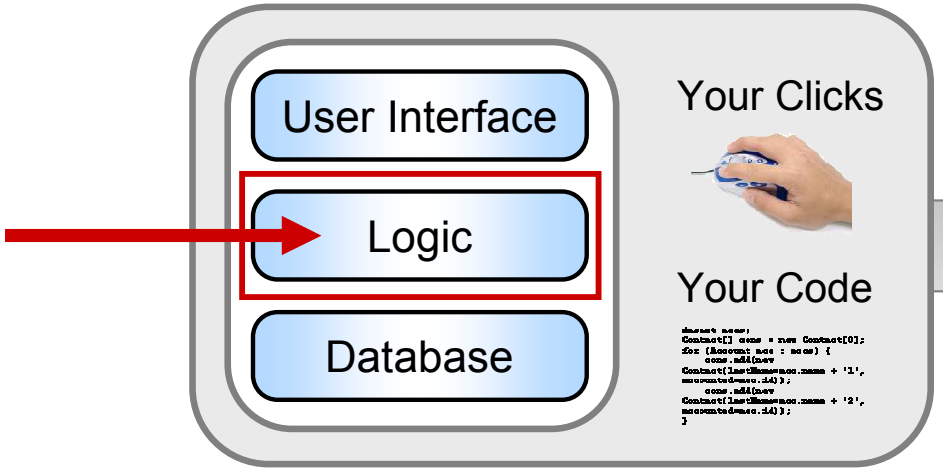




Single tenancy gives each customer a dedicated software stack – and each layer in each stack still requires configuration, monitoring, upgrades, security updates, patches, tuning and disaster recovery.

On a multi-tenant platform, all applications run in a single logical environment: faster, more secure, more available, automatically upgraded and maintained. Any improvement appears to all customers at once.

The Technical Part: Why multi-tenancy matters



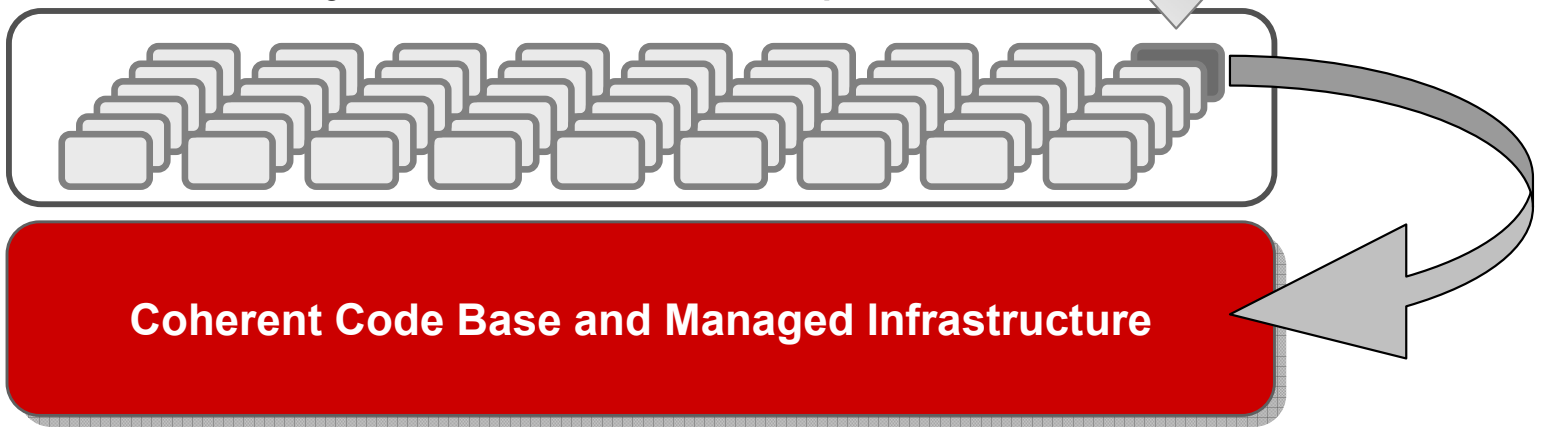
Build *strategic* applications

Customize *any* aspect

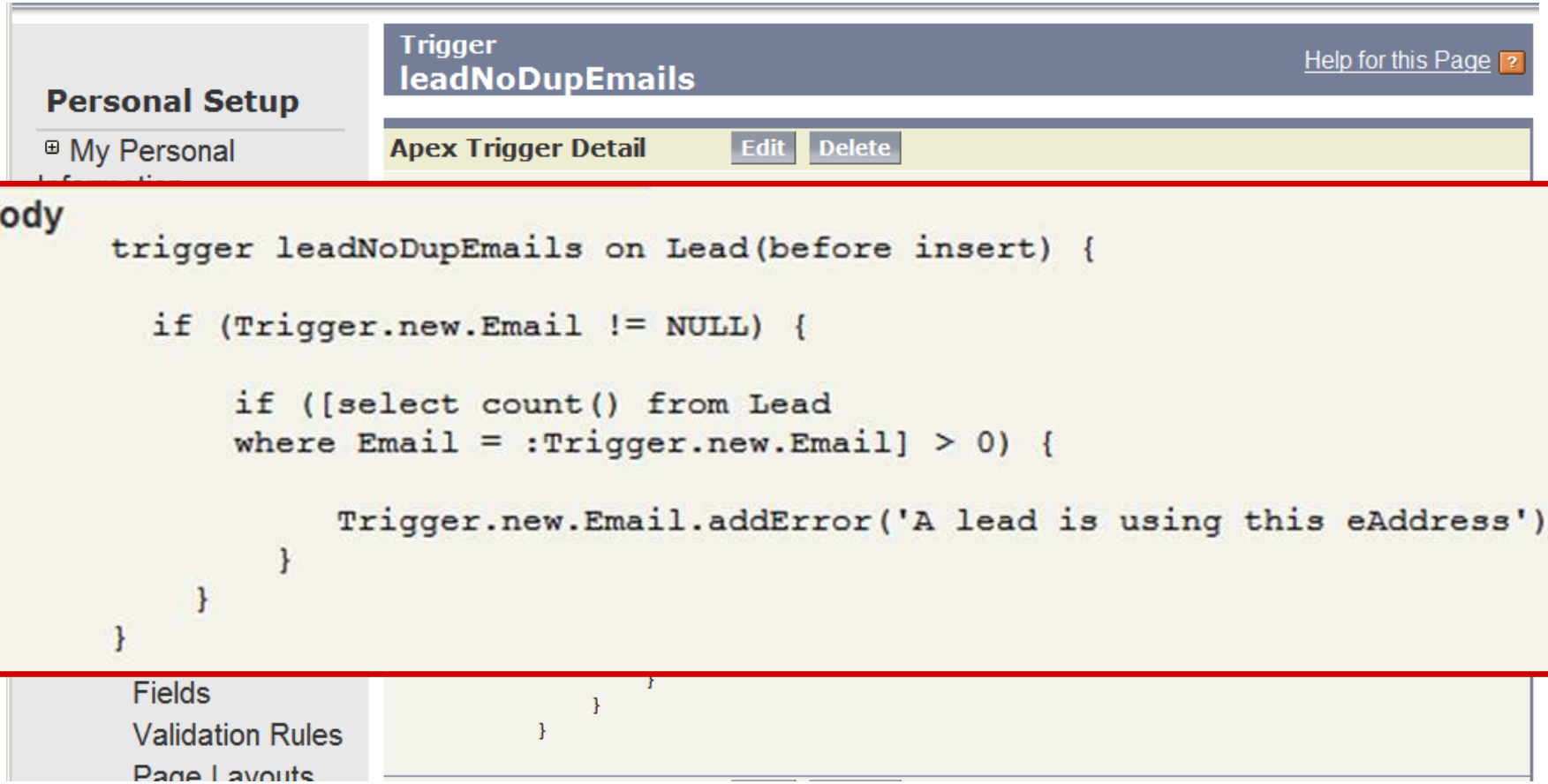
Upgrade when convenient

Retain IP ownership

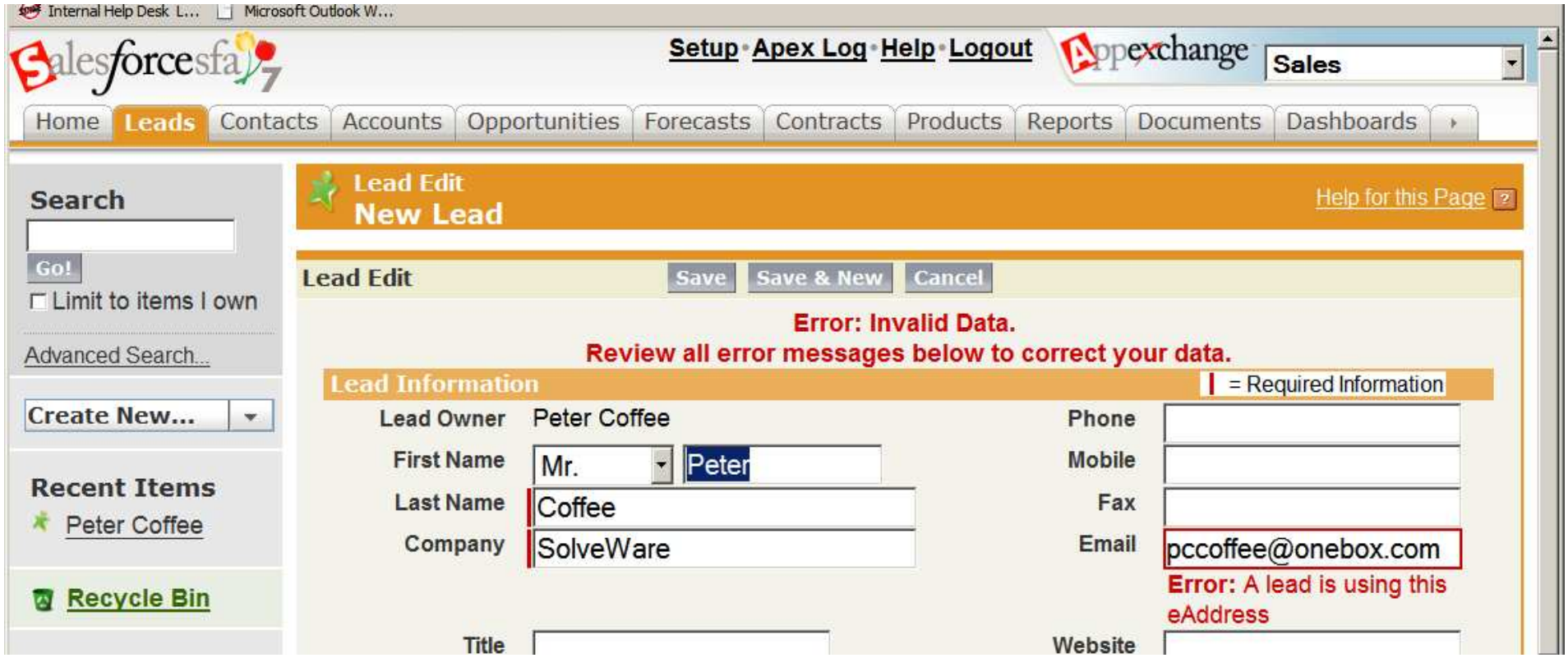
Metadata representations:
Partitioned data, logic and customizations for **multiple customers**



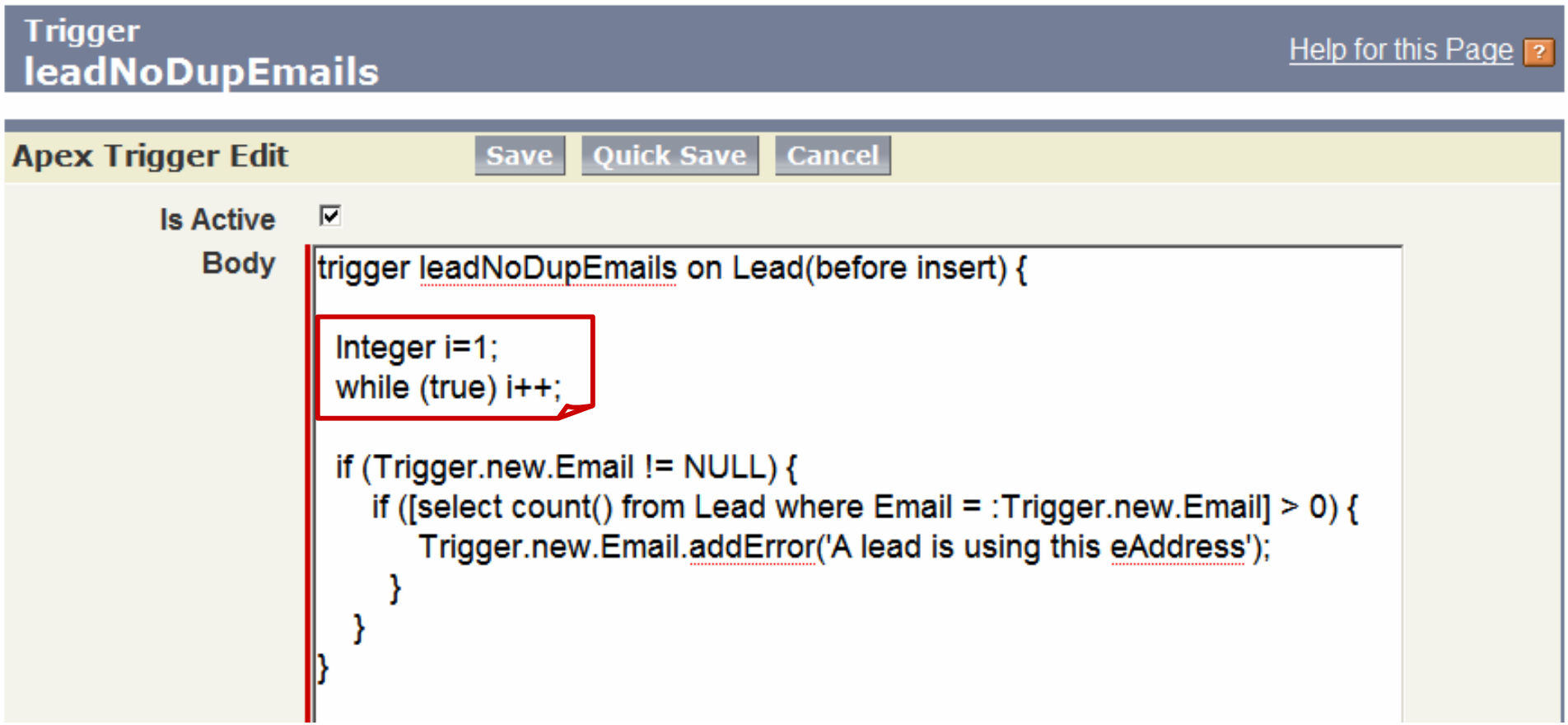
Procedural Power



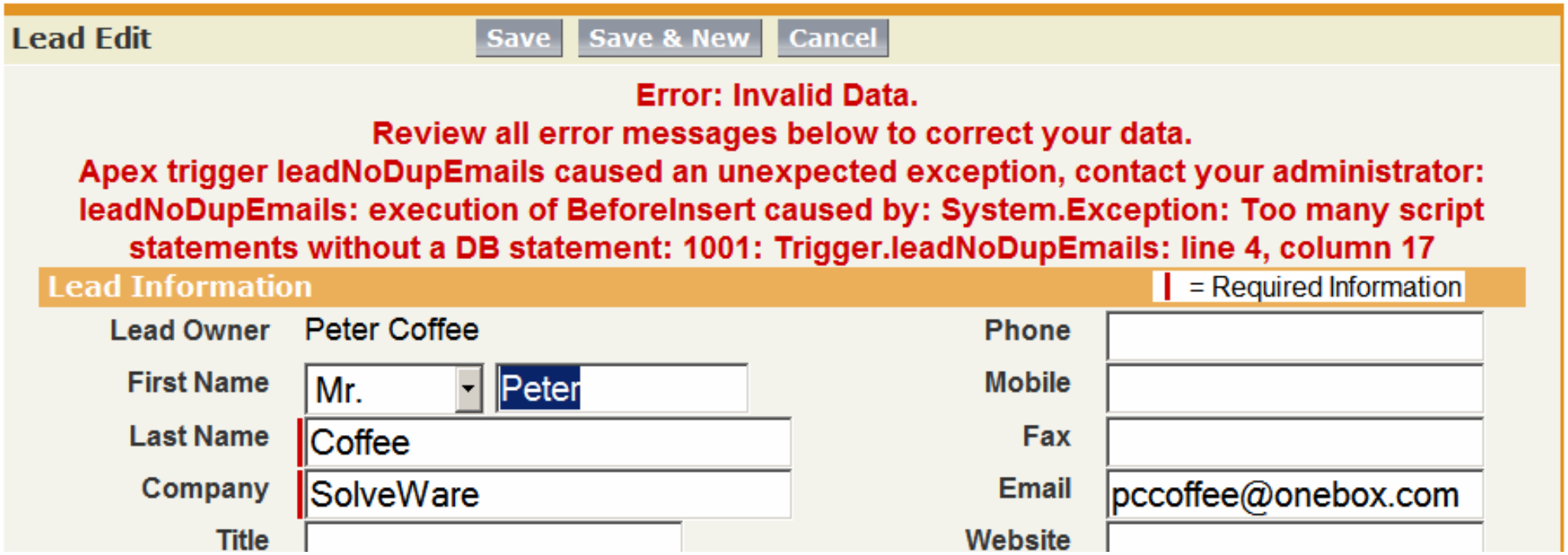
Platform Leverage



Run-Time Governance



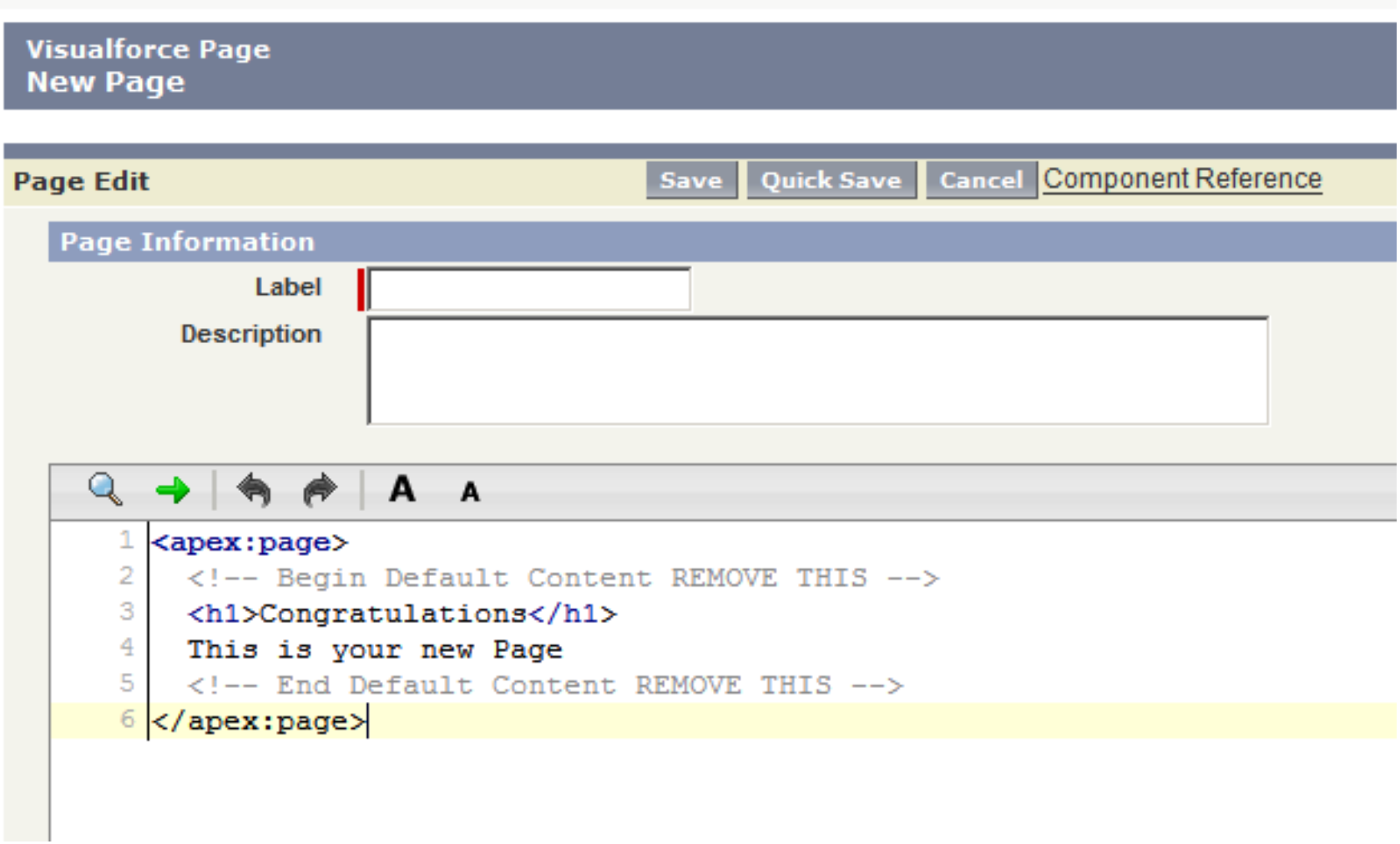
The Platform can be Proactive



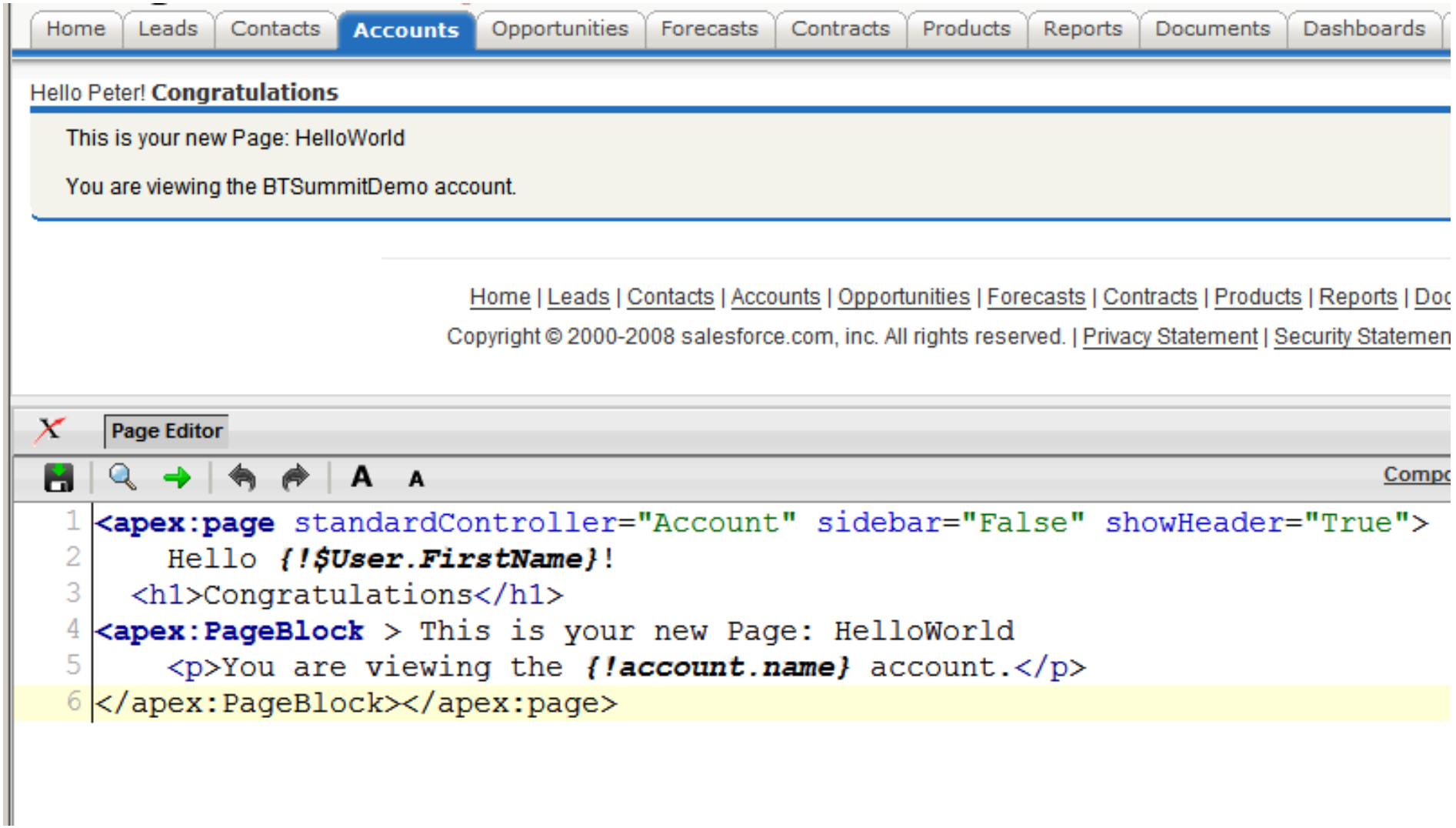
The Developer can take Precautions

```
if (updatedContacts.size() + Limits.getDMLRows() > Limits.getLimitDMLRows()) {
        if ( Trigger.new.size() == 1) {
                Trigger.new[0].addError('You are attempting
                to update the addresses of an account
                with too many contacts.');
        } else {
        for (Account a: Trigger.new) {
                a.addError('You are attempting
                to update the addresses of too many accounts
                at once. Please try again with fewer accounts.');
                }
        }
}
```

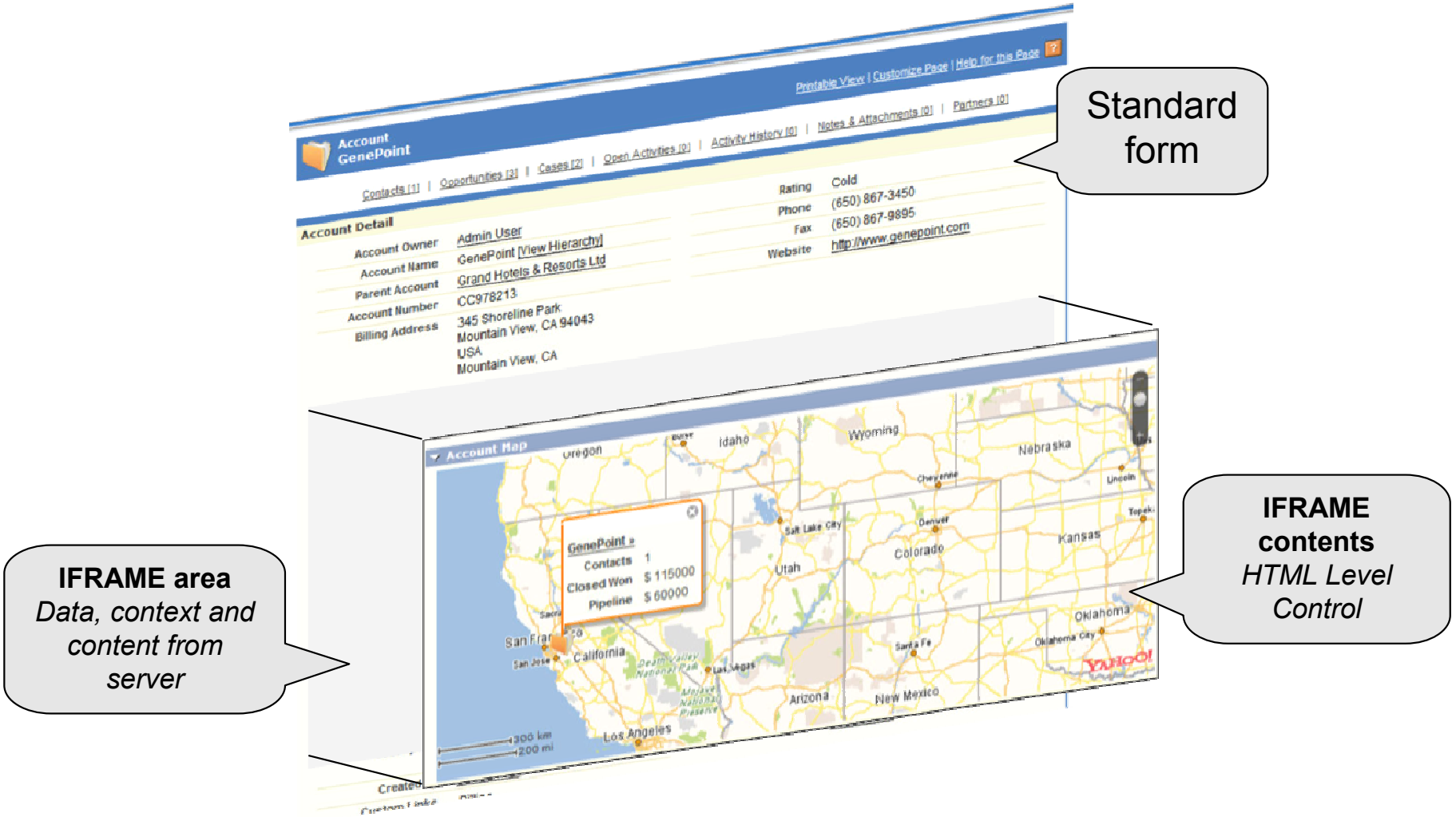

User Interface: Declarative Directness



User Interface: Behind the Glass



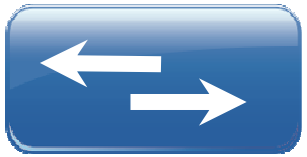
User Interface: Open to Extension & Integration



Development as a Service

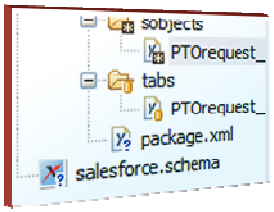
Pioneering Cloud-based Tools and Communities

Metadata API



Easy **Access** to Code and Schema

Force.com IDE

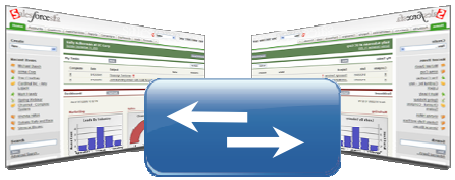


Everything You Need to Build Apps

Force.com Sandbox

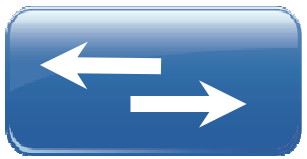
Instantly **Set Up** Dev Environments

Force.com Code Share



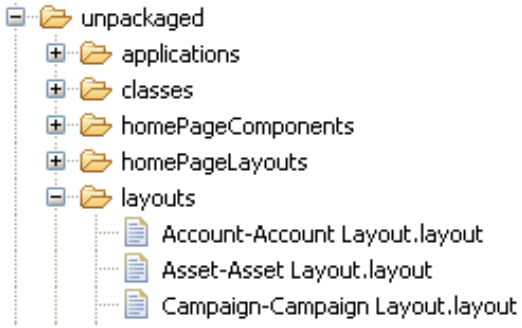
Easy to **Collaborate** on Projects

Metadata API



Easy Access to Code and Schema

Saving Time with the Metadata API



We've done about 20 Salesforce implementations, and have 5 more going right now. We've developed a base template–a set of objects, fields, reports, dashboards, workflow, scontrols, visualforce, and apex that we start with for each client. We then customize that to their needs.

When I first started this work, it would take about a week to get a base install set up. All the objects had to be created in the web interface. Fields and values had to be done by hand. Code had to be copied by hand. It was a drag.

Then came the Appexchange, and we got to package up some things and install them with a few clicks. Then the Appexchange got better, allowing for more things to be packaged.

The last time I did a base install of our template, it took about 8 hours.

Much of this time was getting the page layouts to look right. If you've ever worked on a page layout, you know how complex they are. First you have to get all the fields in the right place by dragging and dropping them. But you then have to click into each field if you want to make it read only. You have to pick the buttons you want to show up on the page. You have to pick the related lists, and then each related list has a list of fields in a specific order, as well as sort order.

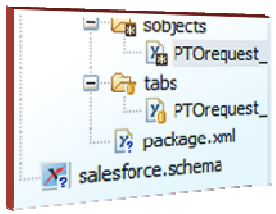
Page layouts were a significant piece of work, and I'd often find out in training with the client that one related list didn't have the right fields, or wasn't sorted the correct way.

The Metadata API and the Force.com IDE have changed how we deal with page layouts. We can now set them up in our template instance and copy and paste them into our customer org.

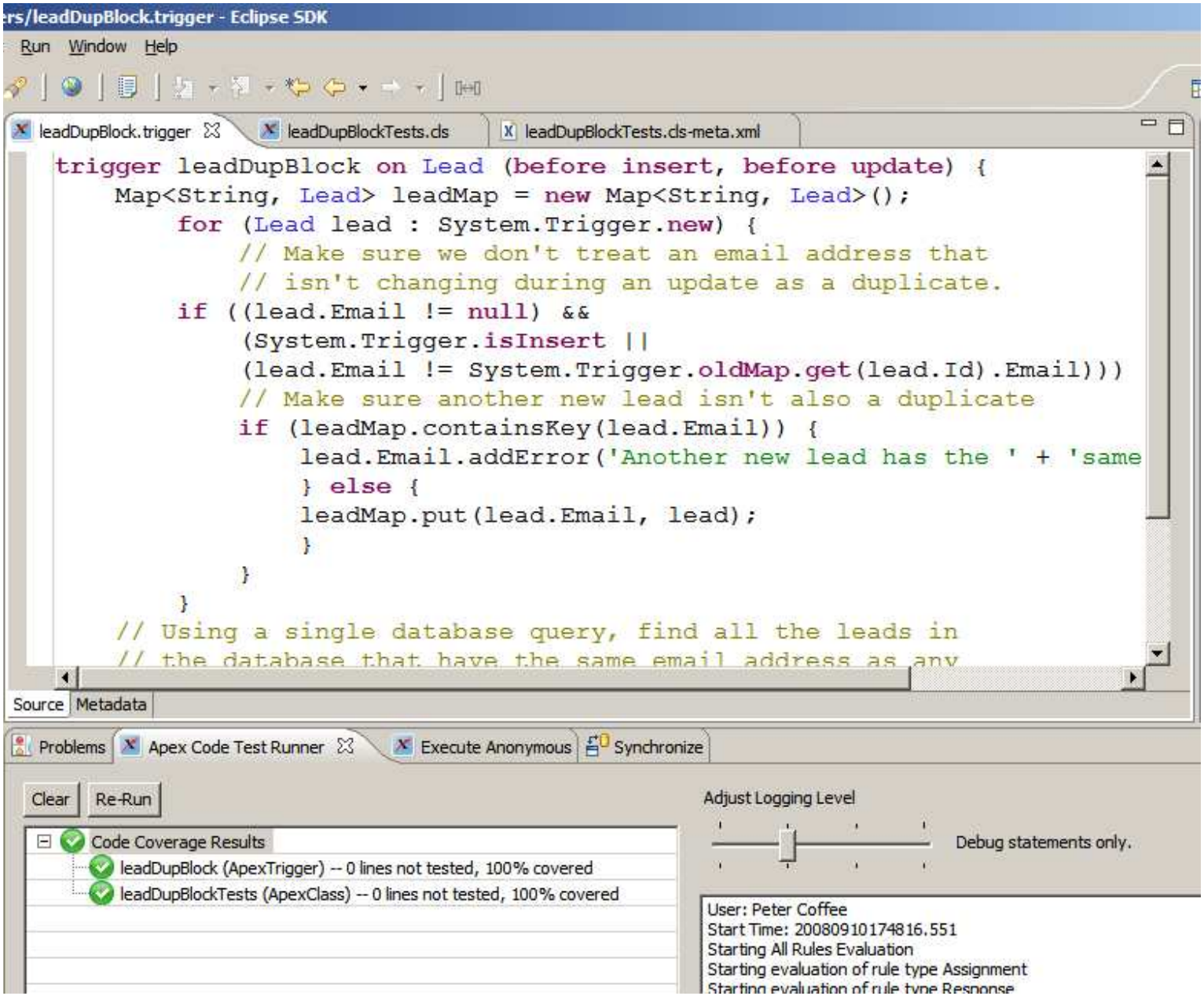
I have a new customer and I set them up with our base template yesterday. It took me 2 hours.

For each customer I deal with, the Metadata API will save me 6 hours of point and clicking in page layouts. I can't tell you how happy that makes me–hand-tweaking page layouts isn't the greatest value-add to the environmental cause.

Force.com IDE



Everything You Need to Build Apps



What's In It for the Developer

- 20-month study of Force.com productivity conducted by Galorath Inc. during 2007-2008
- Work product: calibration of the Galorath SEER cost estimation tool for budgeting of Force.com projects
- Performed under contract to BAE Systems plc to support BAE proposal to FAA

- **Conclusions** (vs. Java):
  - **Requirements definition time reduced 25%** due to rapid update cycle of metadata-defined applications
  - **Testing effort reduced by more than 10%** due to extensive re-use of already-proven code
  - **Development productivity of new code 5x greater**
  - **Overall project cost 30-40% less**

Real-World Results: Professional Services

- Animators at Law, a leading provider of litigation graphics, litigation consulting & litigation technology for many of the largest law firms, pioneered a unique system for identifying the litigation activities of law firms and corporations and **wanted to make the data available to third parties** through a subscription-based service.
- In just a few months – with **no added development staff** – the team created LawProspector, the first comprehensive sales lead and litigation market intelligence tool. The application, built on the Force.com platform, integrates with Salesforce CRM Enterprise Edition and Salesforce CRM Partner Networks.
- LawProspector is integrated with applications from the Force.com AppExchange. **LinkedIn** for Salesforce, enables users to access LinkedIn information directly from Salesforce CRM contact and account records, and **Account News Feed powered by Google News** displays relevant news items from Google alongside Salesforce CRM records.

PaaS Taxonomy: Proliferating Platforms

"Servers as a Service"

Virtual Servers

Database as a Service

Infrastructure as a Service

~Familiar Developer Model
Rapid Scalability

PaaS for the Inquiring Developer

Python App Server

Database as a Service

Infrastructure as a Service

β Offering
Innovative Technology

PaaS as an Application Framework

UI as a Service

Logic as a Service

Integration as a Service

Database as a Service

Infrastructure as a Service

Supports Large-Scale SaaS
Deep-Dyed Multitenancy

Force.com ⇔ Amazon Web Services

Combine cloud infrastructure capability with application platform leverage

Develop in Java, Ruby on Rails, LAMP Stack

Access Mega Storage from Amazon S3

Burst a Force.com App to Amazon EC2

Force.com ⇔ Google App Engine

- Python library and test harness
- Access Force.com Web Services API from within Google App Engine applications

Force.com ⇔ Facebook

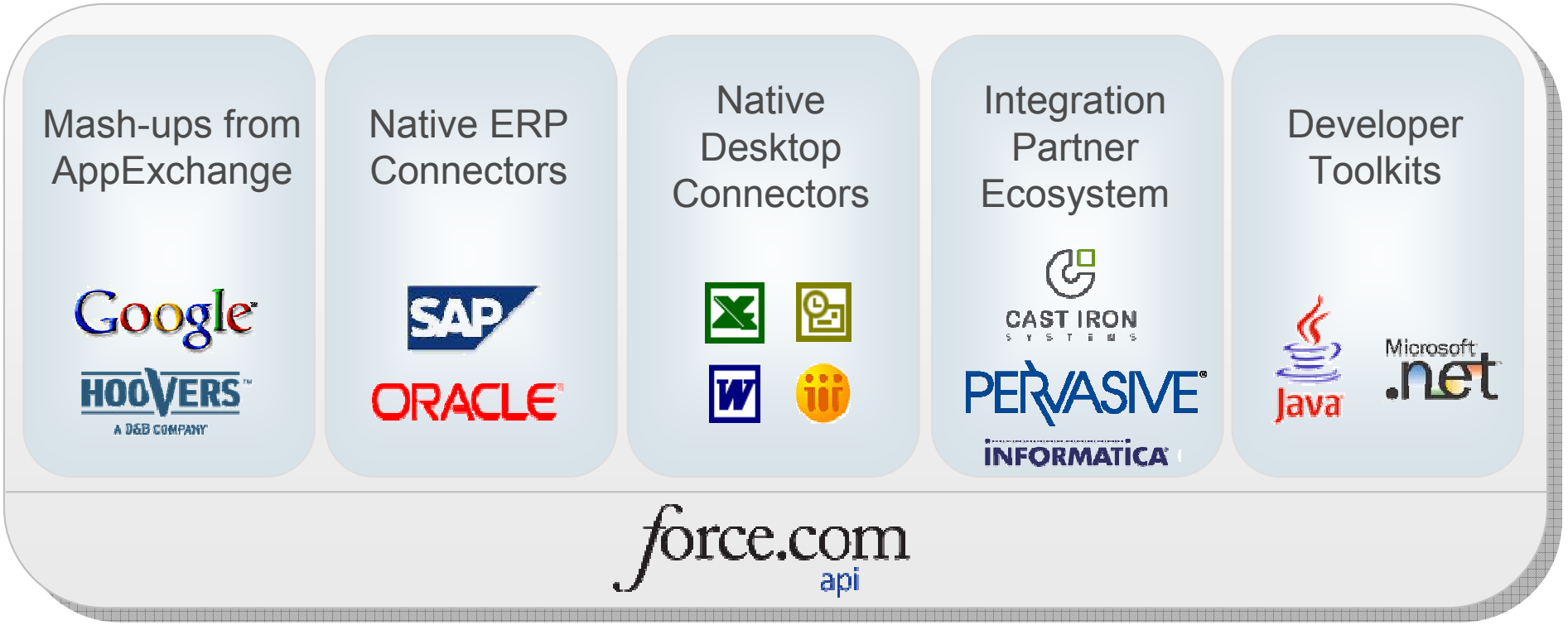
Build enterprise applications with social network outreach

Provide a scalable, cloud-based infrastructure accessible by Facebook applications

The Cloud is a Services Supermarket

Combine platforms: Combine strengths

Leverage from *all* Assets: Integration as a Service



"Apex Connect provides a set of integration technologies that make it possible to bring together multi-tenant platforms, **reduce integration complexity, and improve time to value.**"

— Program Manager, Enterasys

Real-World Results: Health Care

- CRC Health—the nation's largest provider of drug and alcohol treatment services—acquired the country's largest youth treatment provider. The combined organization required a platform to mange patient intake, track Web entities, and streamline operations to increase revenue.
- The company used ACT!, spreadsheets, and other proprietary systems to manage extensive patient data. Only one call center operator could open the spreadsheet at a time, making the process **inefficient, opaque, and unscalable**.
- The company developed a customized user interface on Force.com for 12 users. With help from salesforce.com partner Appirio, CRC Health extended the application to broadly leverage the platform.
- Security levels are matched to what's required to **comply with HIPAA** and other industry regulations. Open APIs enable **tight integration** with legacy tracking systems, Microsoft Outlook, eFax, and other third party apps. Web marketing **effectiveness tracking** within Salesforce CRM indicates to the dollar what is performing and what is not.

APPIRIO

CRC HEALTH GROUP

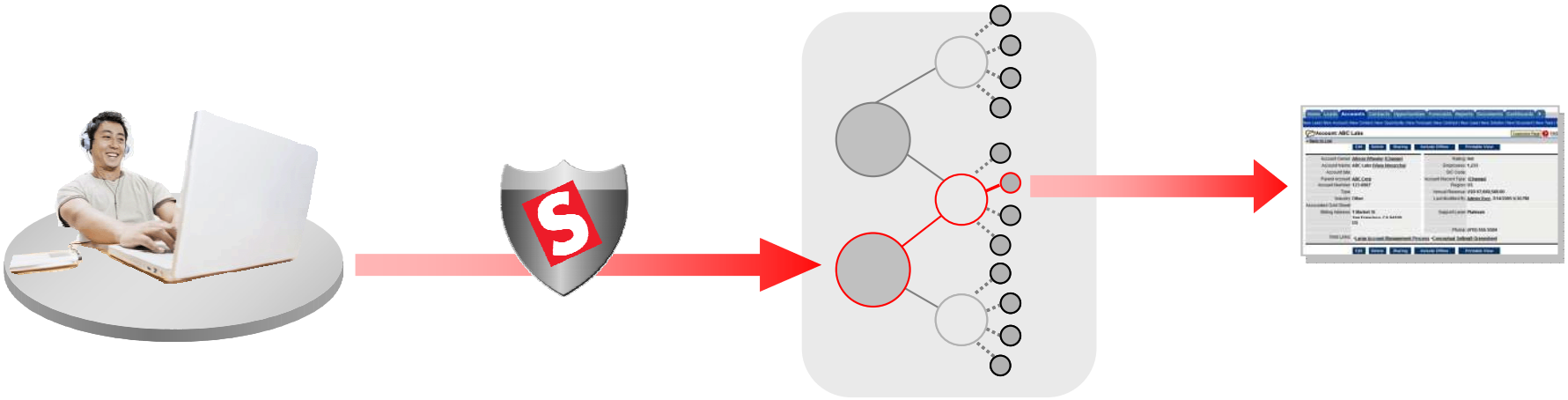
Multi-Tenant Application Security

Login…

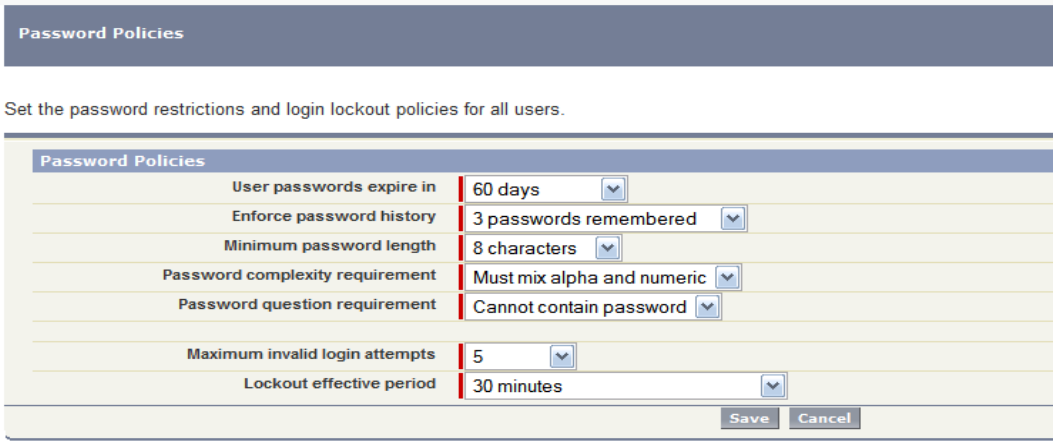
Authenticate…

Apply Data Security Rules…

View Filtered Content



- Password security policies
- Rich Sharing Rules
- User Profiles
- SSO/2-factor solutions



Multi-Tenant Application Security

Strong Session Management

Every row in the database contains an ORG_ID - Unique encoded string

Session Tokens – user unique, non-predictable long random value generated for each session combined with a routing "hint" and checksum, base64 encoded

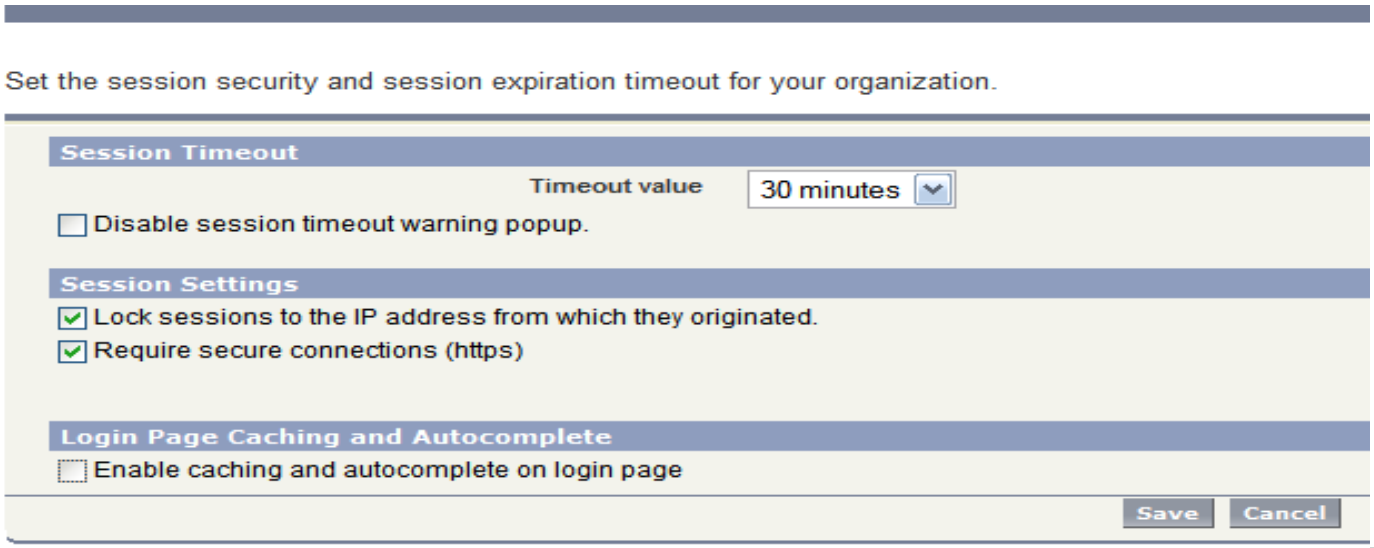
Contains no user-identifiable information

Session Timeout – 15 Mins to 8 Hrs

Lock Sessions to IP – prevent hijacking and replay attacks

SSLv3/TLS used to prevent token capture / session hijacking

Session Logout – Explicitly expire and destroy the session



Multi-Tenant Application Security

- Don't Expect to Make Water Run Uphill
  - Users are easier to crack than protocols…

    …so…
  - Restrict allowable IP addresses
  - Shorten timeout thresholds

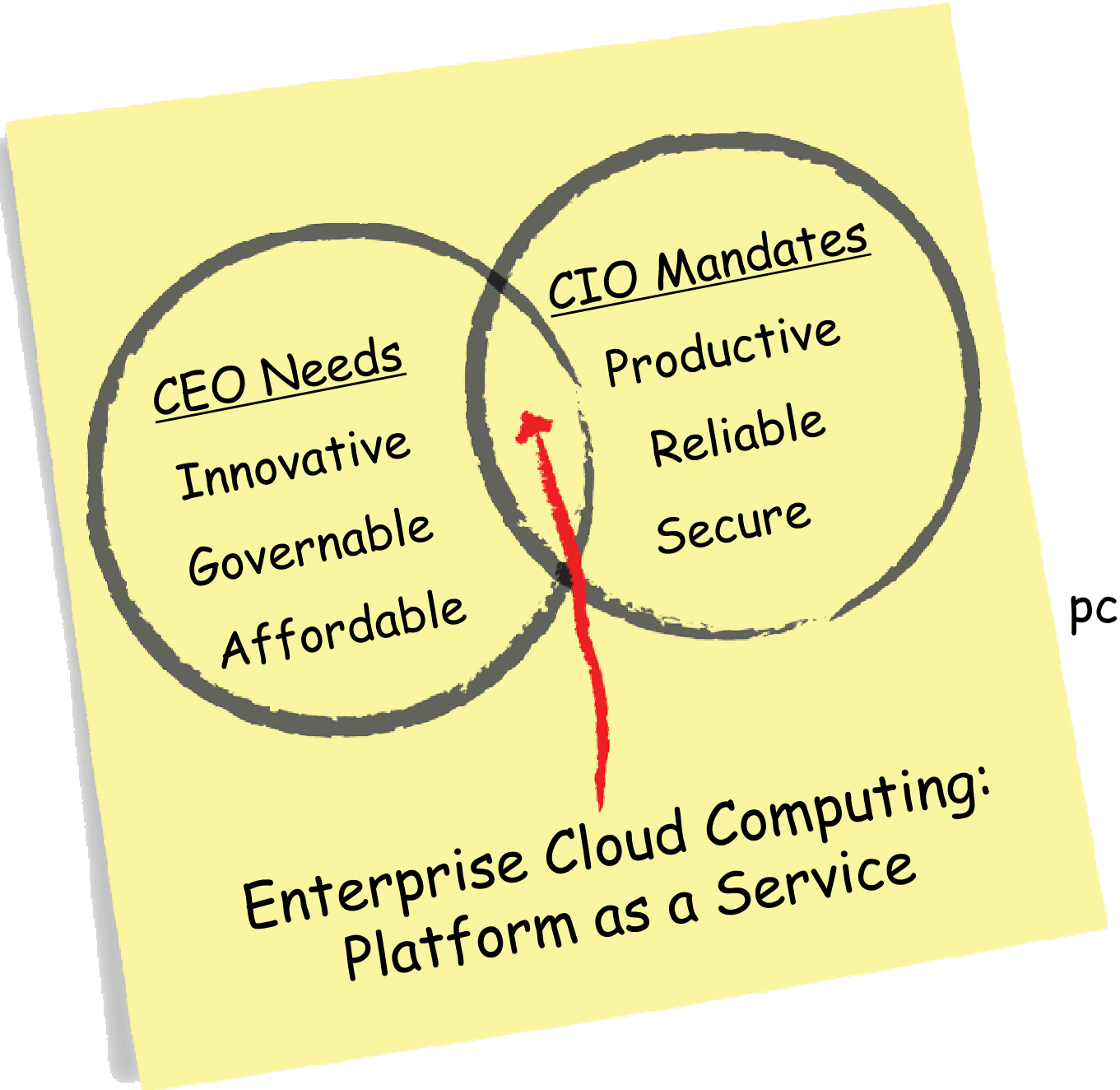
RESTRICT IP ADDRESS IN SALESFORCE

- Balance Capability Against Control
  - For example, encrypted fields (salesforce.com)
  - Only visible to users with "View encrypted data" permission…

    …but…
  - Encrypted custom fields cannot be unique, an external ID, or have default values
  - Encrypted fields are not available for use in filters such as list views, reports, roll-up summary fields
  - Encrypted fields cannot be used to define report criteria, but can be included in report results

Best Practices and Pitfalls

- Adopting the Cloud does *not* mean starting over
  - Retain what's working: **innovate and add value** at Web speed
  - Don't settle for the least unsatisfactory solution:
    treat the Cloud as **a supermarket of services**
- Preserving familiar pain is *not* a measure of success
  - Moving existing complexity into the Cloud avoids short-term pain
  - Mastering new developer models is **a high-return investment**
- *Don't* apologize for doing what made sense two years ago
  - Bandwidth has grown
  - Customizability has grown
  - Costs of doing things the old way are skyrocketing
- **Don't mistake the consumer Web for the enterprise cloud**
  - **Expect high availability and robust security**
  - **Spell out details of data ownership and protection**



Thank you

pcoffee@salesforce.com