



Sink or Swim: Key Technical Considerations of Building and Operating a SaaS Application

March 19, 2009

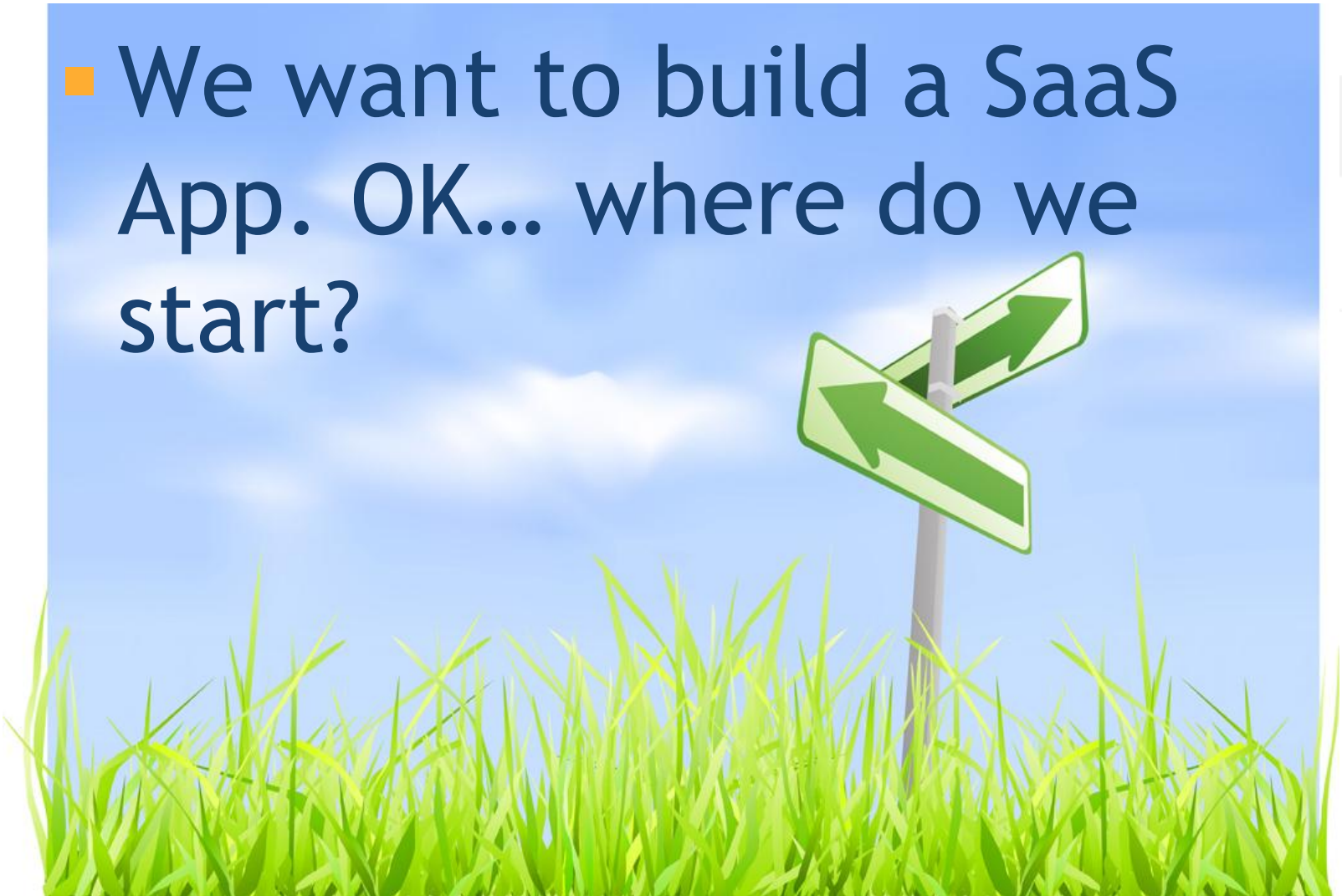
Luis Aburto, CEO - Scio Consulting

Jesse Kliza, Director of Business
Development - Apprenda

Who is Scio Consulting?

- **SaaS Enablement Professionals**
 - » **SaaS Business and Technical Consulting**
 - » **SaaS Product Development Services**
 - » **SaaS Infrastructure Mgmt and Operations**

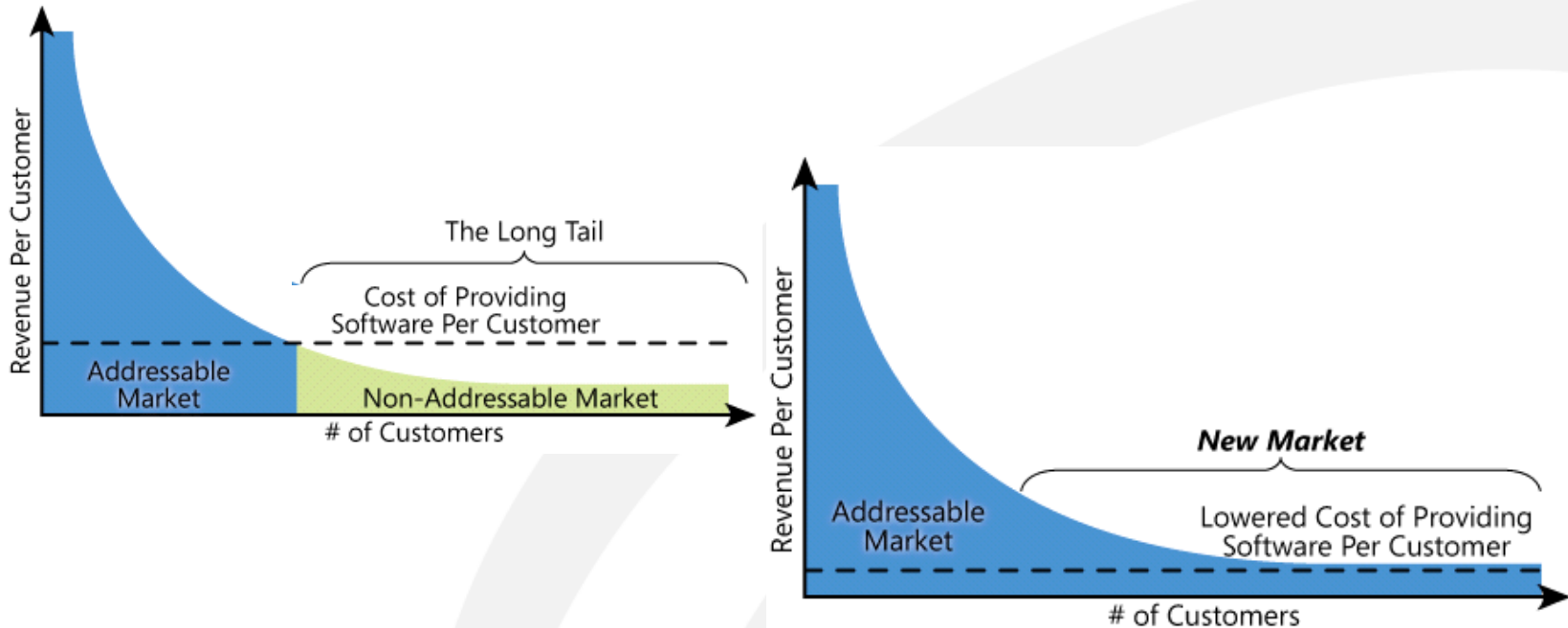
- We want to build a SaaS App. OK... where do we start?



Selecting the Feature Set

- Is there an existing on-premise version of the app?
- Who is the target customer for the SaaS app?
 - » Same as for on-premise?
 - » New target?
- What is the purpose of creating the SaaS app?
 - » Offers opportunity to enter new markets?
 - » Stop losing clients?
 - » Expand reach?
- Consider
 - » Business analytics and metrics
 - » Mobile devices

- Develop for the Long Tail



- 80-20 Rule
 - » 20% of the features provide the 80% of the value for end users

Skills for Building

- » Product Management
- » Web Architecture & Development
- » Web/RIA UI Design
- » Infrastructure Architecture
- » Web Testing

Skills for Operating

- » Web-based marketing & sales
- » Infrastructure management
- » Web application management & performance monitoring
- » Web-based customer service and technical support

Defining the right level of “SaaSification”

“SaaSification” Levels

Characteristic	Hosted (ASP Model)	Application Virtualization Approach	Full SaaS Approach
Tenancy Model	Single Tenant	Single Tenant with Tenant Mgmt. Tools	Multi-tenant
Modification Required to Existing App	Minimal	Minimal	Moderate to Extensive
Operative Overhead	Highest	Moderate	Lowest
Scalability of the Approach	Lowest	Low	Highest
Initial Investment	Lowest	Moderate	Highest

- Application Virtualization tools

- » Wrapped Apps



- » Parallels



- » Citrix



- » Etc.

- Considerations

- » Is there legacy code that could be leveraged or that would be difficult to re-write?

- » How many new implementations per year are forecasted?

- » Has the SaaS model been proven in the target market?

- Multi-tenancy
- Scalability
 - » Load balancing
 - » Routing
- Availability
- Performance
- Configuration-driven Customization
 - » look and feel
 - » Functionality
- Integration
- Security
 - » Identity management
- Usability
- Communication
 - » e-mail, sms, etc.
- Globalization
- Audit and compliance
- Backup and recovery

These factors influence **App Architecture** as well as **Infrastructure Architecture**

SaaS-specific functionality that must be present in a SaaS application

- Pricing Engine
- Billing Engine & Payment processing
- Tenant and Subscription Management
- Service Provisioning
- Usage and Performance Monitoring
- Subscriber Management & Self-Service

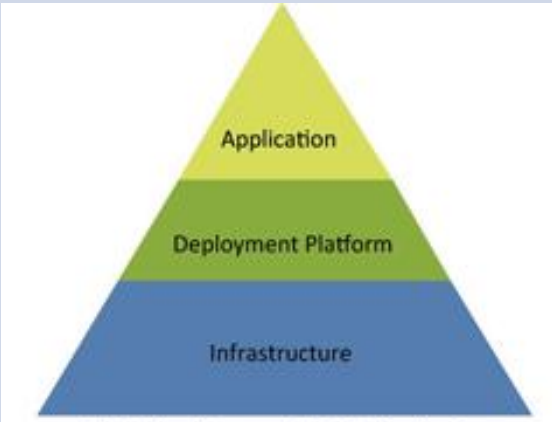
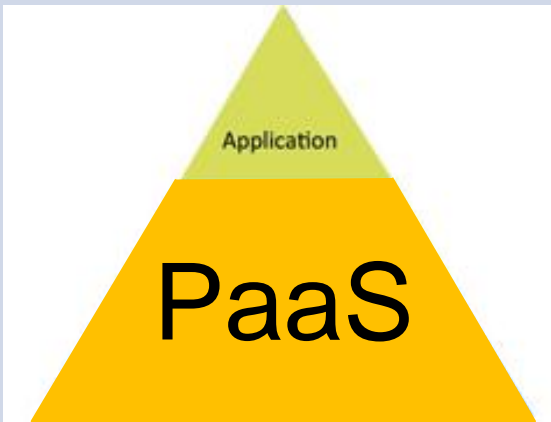
**This functionality takes from 20 to 50% of
The development effort of a SaaS App**

Leveraging commercial SaaS components and services to expedite development

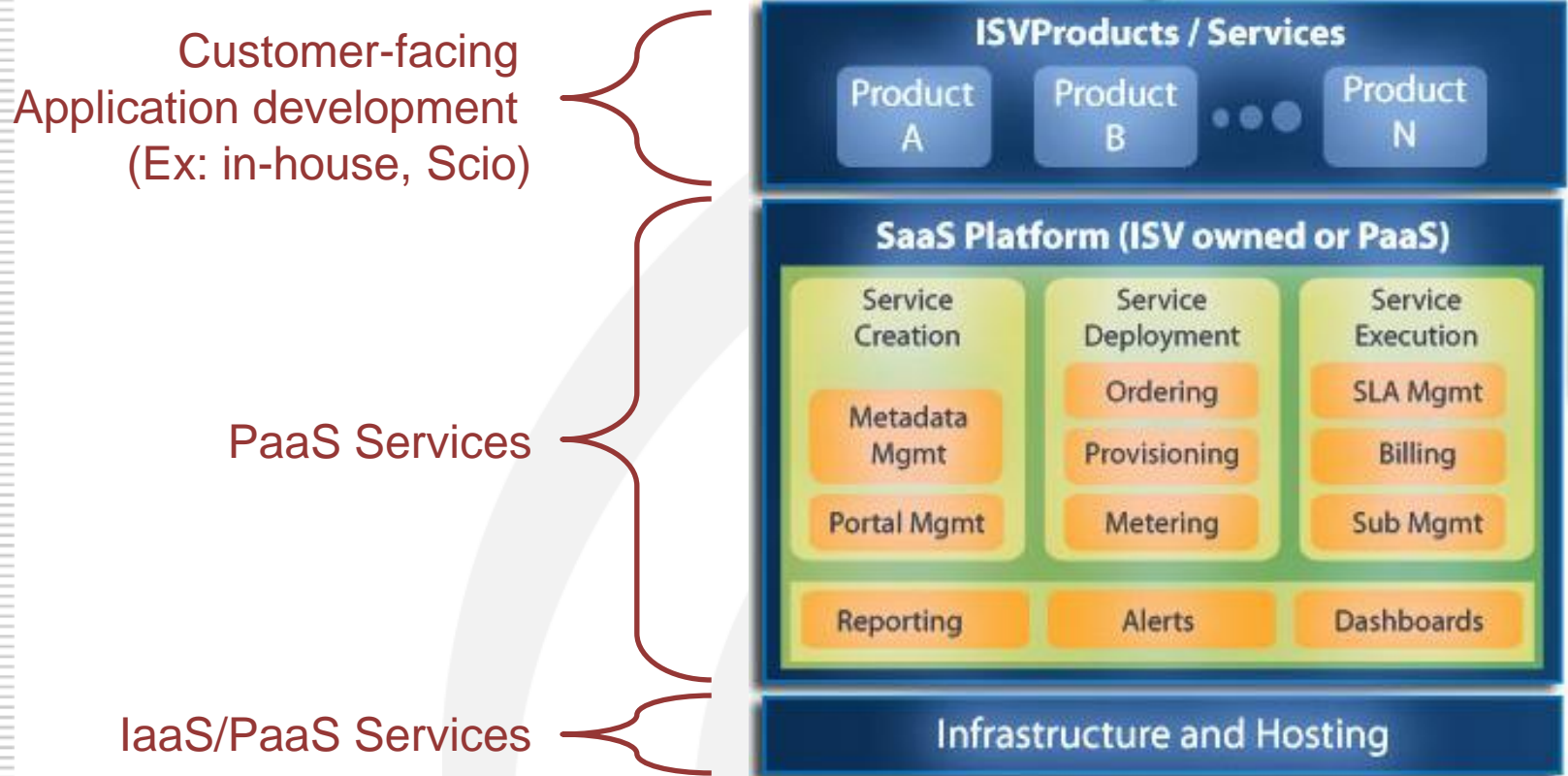
- Billing
 - » Zuora, Vindicia
- Customer Management
 - » Aria Systems
- Payment Processing
 - » Paypal, etc.
- Monitoring
 - » Absolute Performance, TrustSaaS, SaaSMonitor.com
- Integration
 - » Boomi, Cast Iron Systems
- Analytics
 - » Sonoa



Choosing a technology stack and/or PaaS

From the Ground Up	Through a PaaS (Platform as a Service)
 <p>Technical Layers of SaaS Applications</p>	 <p>Technical Layers of SaaS Applications</p>
<p>Platform Examples:</p> <ul style="list-style-type: none"> • LAMP • Win, IIS, ASP.NET, SQL Server • Ruby on Rails 	<p>PaaS Examples:</p> <ul style="list-style-type: none"> • SaaSGrid • Force.com • Google App Engine, Bungee Lab
<p>Hosting Alternatives:</p> <ul style="list-style-type: none"> • In-house • Co-location • Cloud Computing (EC2, GoGrid) 	<p>Hosting:</p> <ul style="list-style-type: none"> • Included

SaaS Technology Stack Details



- **In-house Datacenter**

- **Co-location**

- » Managed Hosting
 - Peer1, Rackspace, Bluelock, etc.
- » Un-managed Hosting
 - ServerBeach, The Planet, etc.
- » Value-added Managed Hosting
 - OpSource

- **Cloud Computing**

- » Amazon EC2, GoGrid, Mosso, etc.

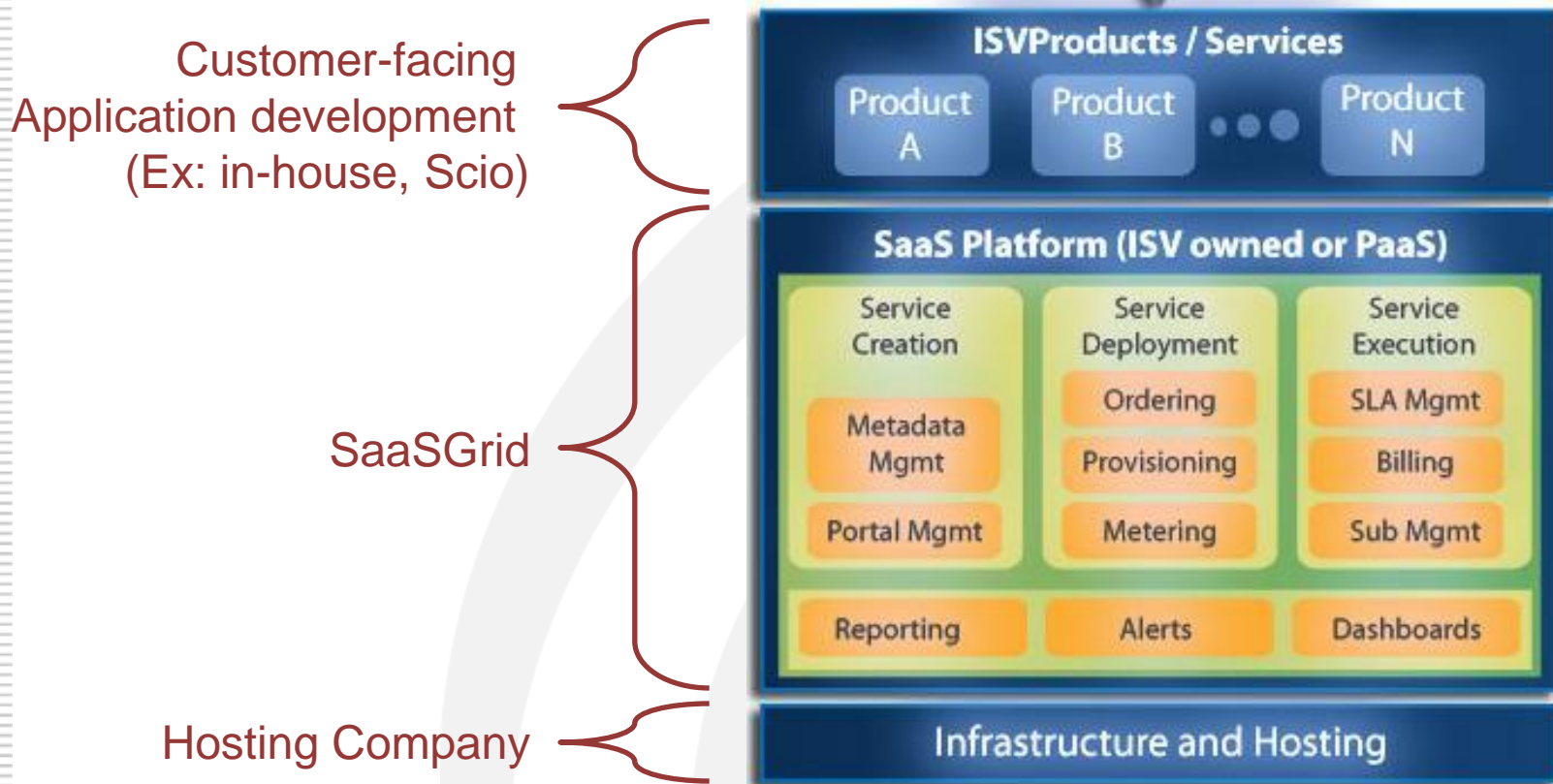
- **Included with PaaS**

- » SaaSGrid, Force.com, etc.



- Testing/QA/QC
- Release cycles
 - » How do we manage maintenance windows?
 - » How do we manage upgrades and new functionality without affecting customers?
- Hosting (Infrastructure)
 - » Performance monitoring
 - » Up-time
- Reliability, replication and recovery
- Compliance and auditing
- Contract management
- Customer Service

SaaSGrid as a PaaS





Jesse Kliza
Director of Business Development
Apprenda

Let's talk about YOU

- New ISVs with a need to focus resources
- Existing ISVs with successful businesses looking to break into the SaaS space
 - Complex and powerful applications
- Skilled engineering professionals
 - Want to leverage your existing skills
 - Concerned about:
 - How to deliver SaaS efficiently
 - How to roll out a single update to all of your customers without downtime
 - Application scalability and performance
 - Infrastructure cost
 - Security
 - Multi-tenancy

Apprenda: “A Team of Patients turned Doctors”

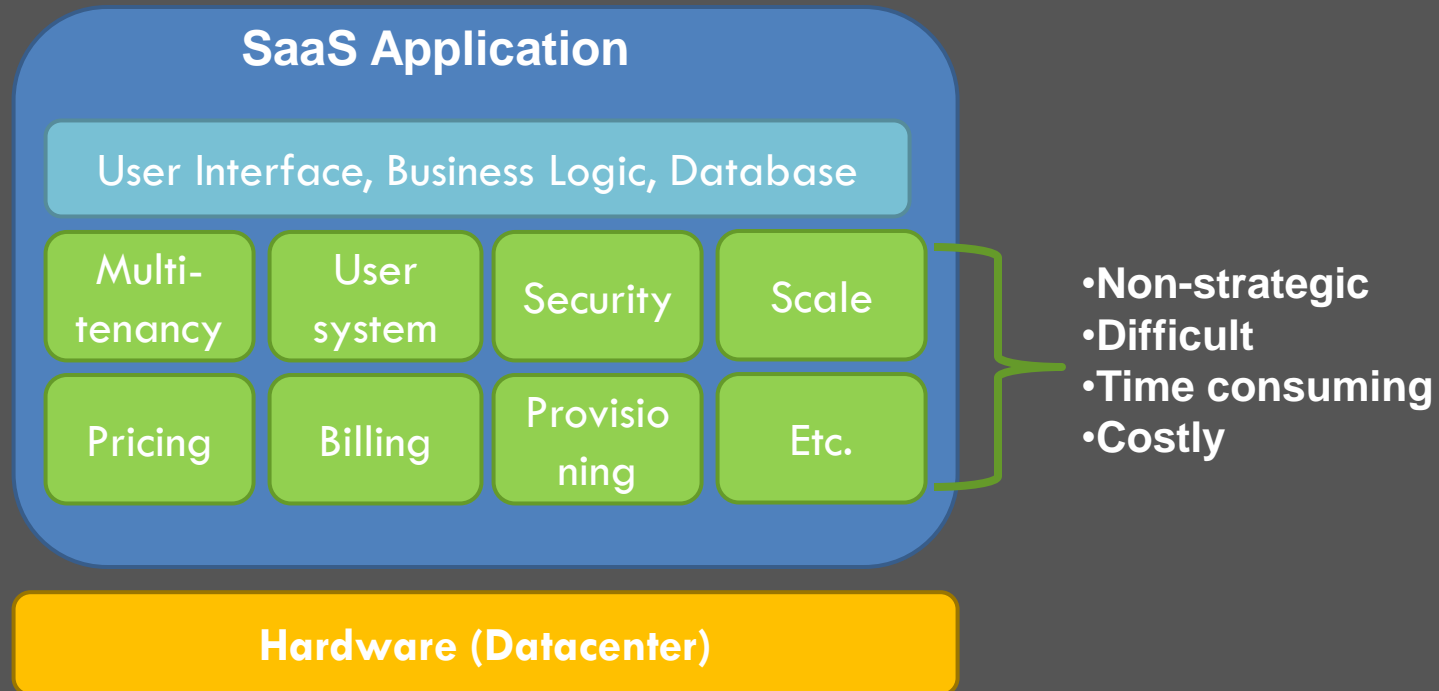
- Understand the “pain” because we’ve been there
- Team members with SaaS experience ranging from software architecture to business modeling.

What is SaaSGrid?

- “Operating System” for SaaS Applications
- Injects “SaaS DNA” into your applications
 - ▣ Convert single tenant → single instance, multi-tenant
 - ▣ Provides all mission critical SaaS functionality
- Drastically simplifies building a SaaS business

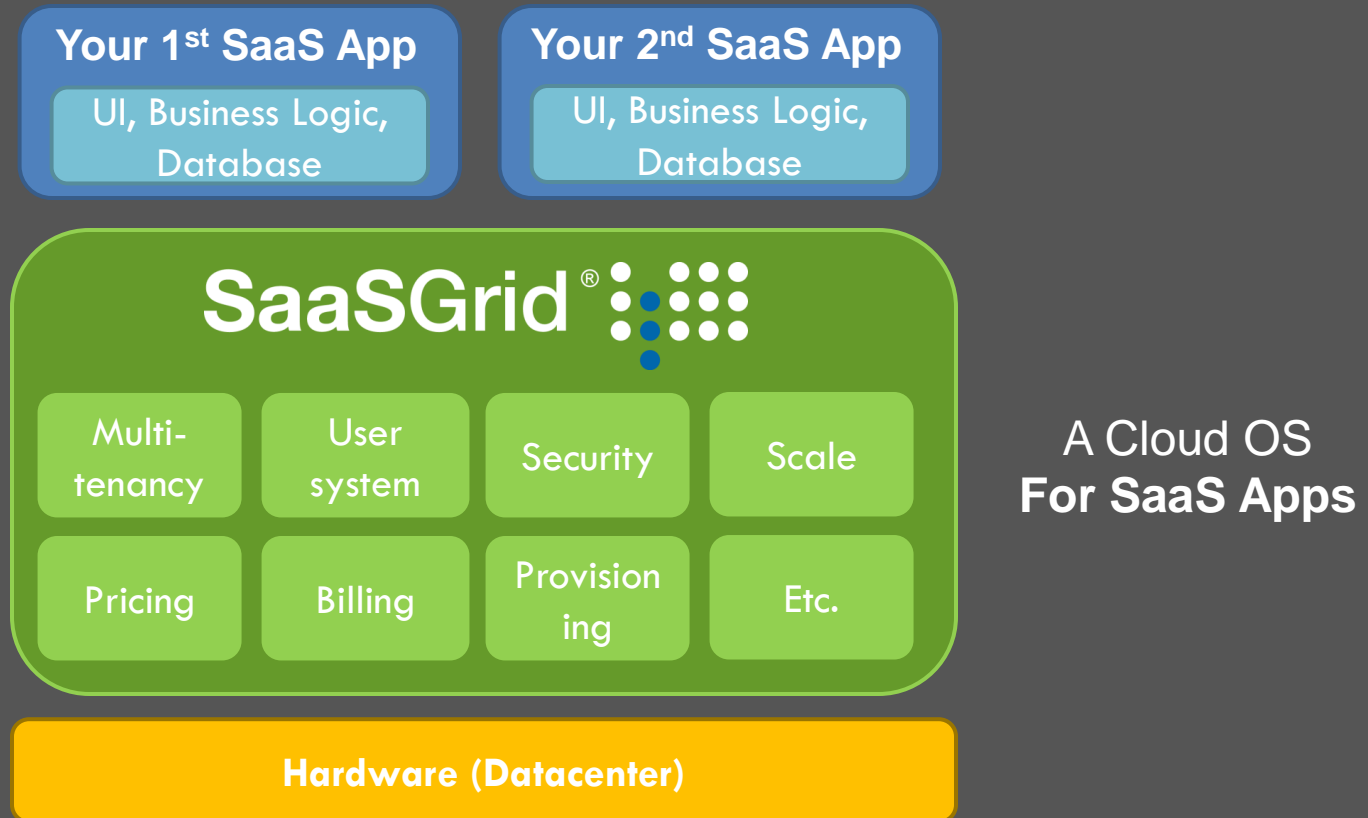
Why “Operating System”?

SaaS Applications Before SaaSGrid



SaaSGrid – The OS for SaaS

SaaS Applications Using SaaSGrid



A Cloud OS
For SaaS Apps

Zero Effort Multi-tenancy

- Apps are written single-tenant, deployed as multi-tenant
 - ▣ Static transformations (e.g. schema transformations)
 - ▣ Runtime instrumentation
- Most Efficient SaaS – single instance, multi-tenant
- Tenancy strategies can differ across the 3 application layers
- Configuration driven - SaaSGrid defers database isolation to deploy time!
 - ▣ Used to be a significant design time decision
 - ▣ **Multi-tenant** → DB is transformed on the fly
 - ▣ **Single-tenant** → A DB is provisioned per customer

Flexible Commercialization

- 5% API driven, 95% portal driven
- Monetization
 - ▣ Define what to charge for and how to bundle
- Provisioning
 - ▣ Get a customer “on board”
- Metering
 - ▣ Track usage so it can be charged for
- Billing
 - ▣ Issue invoices, collect & manage money

Real ISVs don't want to hear...

- “Just throw all of that stuff out, and start from scratch”
- “Here’s a great new programming language that you can learn in your spare time, AND you can only run it in one place...forever.”
- “Here you go little fella...this simple wysiwyg editor should make it easier for you and it’s not TOO constraining.”

SaaSGrid was built for... YOU

- A real web based OS that addresses the core SaaS architecture problems
- Allows you to leverage a known stack (.NET) for application development, and weave in SaaS. Existing assets are important!
- Provides all of the necessary SaaS business tools, hooked deep into the architecture
 - ▣ Allows for very interesting value introduction
- **Decouples hosting from runtime, giving you choice and protection!**

The SaaSGrid Big Picture

- Get to Market Much Faster
- Save Money (Upfront + boost in gross margin)
- Reduce Your Risk Profile
- Focus on Your Customer's Needs
- Allocate resources on strategic initiatives

Key Takeaways

Developing SaaS successfully
involves a lot more than
putting your application
on the Web.

Feature set must be aligned
with vision and strategy
of the new SaaS app

And

Think Web!

Identify gaps in your in-house
skill set and define
how you are going to
fill them.

True Multi-tenant SaaS is ideal

but

in some cases partial/hybrid
solutions may be appropriate

Get your Architecture right
upfront.

Making corrections along the
way will be complex and
expensive.

Creating all the “SaaS plumbing” requires significant effort.

Plan for it.

Hosting alternatives abound.

But not all hosting was born equal.

Go with the highest level of service you can afford.

Key Point #8

- Selecting the right PaaS will
 - » Simplify technical decision process
 - » Accelerate time to market
 - » Reduce development and operating costs

- SaaS Add-ons (monitoring, billing, etc.) can save time and money while adding value to your operations

Conclusions

- SaaS is a different ball game
- There is no magic, just a learning curve
- Build vs Buy applies to expertise, tools, platforms, etc.

- Good luck!



<http://www.sciodev.com>

Luis Aburto, CEO
408.404.3897 x 254
laburto@sciodev.com

<http://www.apprenda.com>

Jesse Kliza, Director of Business Development
518.383.2130 x 306
jkliza@apprenda.com