

Making the Leap: A Report From the Front Lines of Policy Administration System Replacement

ACORD LOMA Systems Forum
Las Vegas, NV
May 14, 2008



Craig Weber
Senior Vice President
cweber@celent.com

About Celent

We are...

The industry's premier provider of

- Business and IT strategy research
- Ongoing advisory services
- Vertically-oriented consulting

An experienced presence in insurance

- Insurance practice began in 2001
- Staffed by industry experts
- 7 FTEs devoted to insurance, with additional coverage by analysts in other practice areas

A truly global firm

- Offices in North America, Europe and Asia
- Reports in English, French, Japanese, Mandarin, and more
- Coverage that spans many geographies

A division of Oliver Wyman

- Global financial services consultancy
- 1,400 consultants, offices in 40 countries
- Domain experts throughout financial services, mirroring Celent's coverage almost perfectly

Agenda

- Introduction
 - Modern vs. Legacy Definitions
 - Performance Comparison
 - Vendor Selection Process
- Tour of Celent PAS Vendor Rating Model
- Tour of Celent Policy Admin Business Case Model
- Conclusions & Discussion

Key Statistics

- New P/C PAS deals, 2007: 170+
- P/C PAS “Upgrade” deals: 90+
- New L/H PAS deals, 2007: 45
- L/H PAS “Upgrade” deals: 30+

Defining Legacy and Modern Systems

AREA	LEGACY	MODERN LEGACY	MODERN
Platforms	<ul style="list-style-type: none"> • Primarily mainframes • Depends on PAS vendor 	<ul style="list-style-type: none"> • Mixed, depending on scale required 	<ul style="list-style-type: none"> • Can be almost anything!
Code Base	<ul style="list-style-type: none"> • The usual suspects: COBOL, Assembler, etc. 	<ul style="list-style-type: none"> • Mix of modern and semi-modern: C++, Visual Basic, PL/SQL, etc. 	<ul style="list-style-type: none"> • Modern languages: .NET, Java, PHP
Databases	<ul style="list-style-type: none"> • Tied to platforms 	<ul style="list-style-type: none"> • Sometimes tied to a single database 	<ul style="list-style-type: none"> • Typically any relational database
Integration Methods	<ul style="list-style-type: none"> • Possibly proprietary • Often wrapped by XML/Web services/SOA 	<ul style="list-style-type: none"> • Varies widely, from limited options to full array of options 	<ul style="list-style-type: none"> • Typically leverages SOA as primary option (in addition to other options)

Modern vs. Legacy Performance

AREA	LEGACY	MODERN
Functionality	<ul style="list-style-type: none"> • Generally extensive functionality developed over many years • Additions to functionality very difficult 	<ul style="list-style-type: none"> • Good, getting better out of the box • Additions to functionality relatively easy, rules externalized from code
Integration	<ul style="list-style-type: none"> • Can be difficult, limited if SOA not enabled • Upgrades required (if available/ possible) to re-orient toward standards-based approaches such as ACORD XML • Often not modular 	<ul style="list-style-type: none"> • Typically easy: systems designed as components, are easy to connect • Based on existing standards, and adaptable to emerging standards

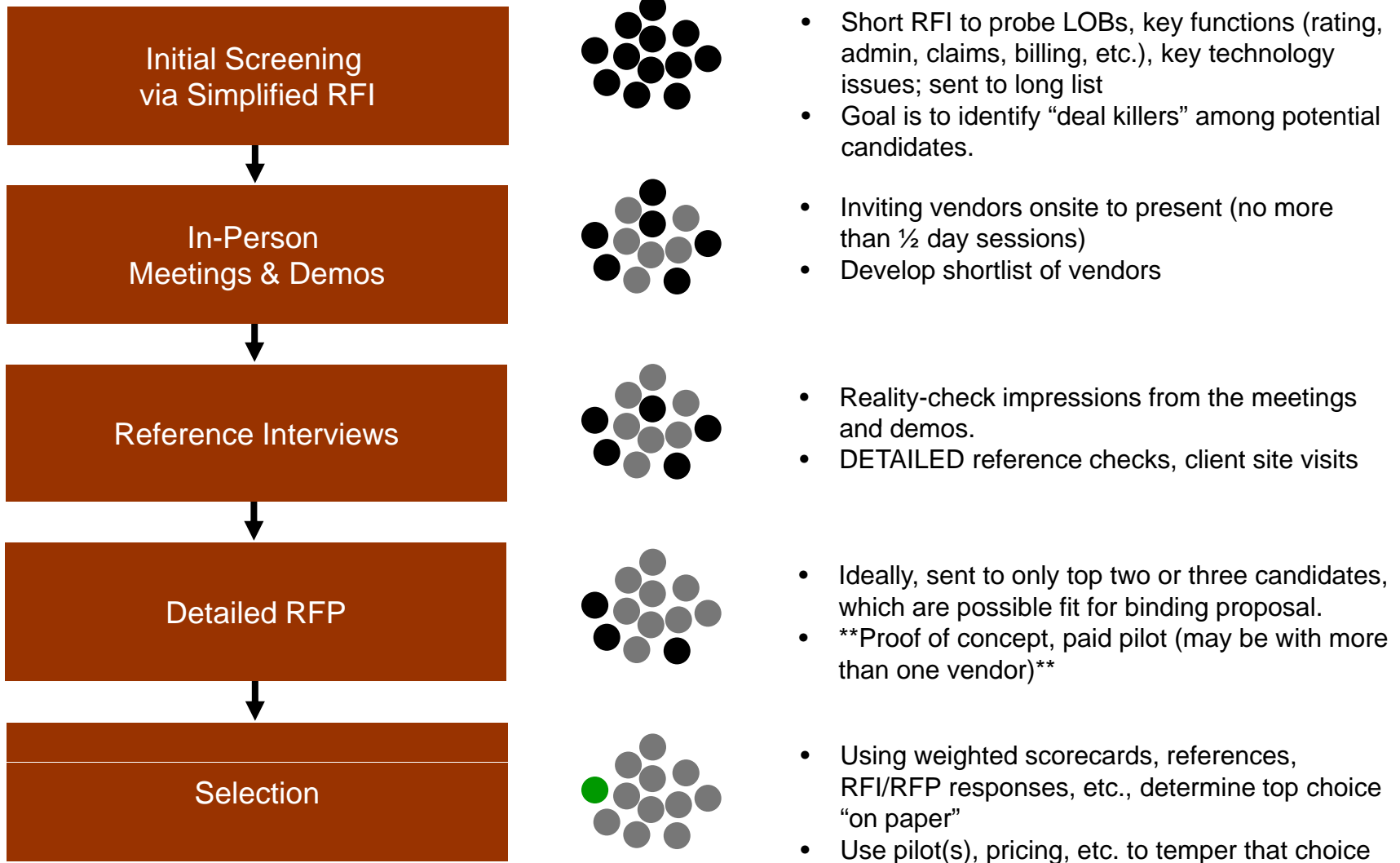
Modern vs. Legacy Performance cont.

AREA	LEGACY	MODERN
Support	<ul style="list-style-type: none"> • Expensive, labor intensive, may be unsupported; may eventually require use of offshore resources • IT-oriented: business users essentially submit requirements to IT 	<ul style="list-style-type: none"> • Cheaper, configuration-based • Business- and IT-oriented: IT becomes more strategic, business becomes more hands-on
New Product Creation	<ul style="list-style-type: none"> • Customizations or clone-based, minimally configurable • Large existing product base • Months for coding, not including testing 	<ul style="list-style-type: none"> • Component based, reusable • Configure, don't code products • Days or weeks to build products
Scalability	<ul style="list-style-type: none"> • Scale achieved at significant cost by upgrading the mainframe(s) 	<ul style="list-style-type: none"> • Depends on platforms and database • But scale may require less computing power (clusters, grids, etc.)

Modern System Rationale

- Faster payback.
- MUCH lower long-term TCO, driven largely by staff savings.
- Flexibility to ease acquisitions, achieve organic growth more easily, and avoid the upgrade void
 - Upgrading via wizards or patches
 - Takes hours or days rather than months
 - Costs in the thousands of dollars, not the hundreds of thousands or millions
- Put business back in hands of the business and let IT focus on innovation for competitive advantage

PAS Vendor Selection Process



Tour of Celent Policy Admin Vendor Rating Model

Tour of Celent Policy Admin Business Case Model

Risk Factors/Unknowns When Choosing Modern System

- **Scalability** of modern systems
 - Mitigation: Scalability testing of proposed system(s)
- **Readiness of all modules** or lines of business in modern systems
 - Mitigation: Plan implementation/deployment around readiness, but see last bullet point!
- Implementing a modern system will result in **massive change**
 - Mitigation: Strong organizational change, training, and communication plans
- **Pace of implementation** should be rapid to reduce risk
 - Mitigation: Directives from the top; buy-in at all levels; authoritative PMO; “drop dead” deadlines
- **If the modern system won't scale or be ready, then consider waiting!**

Conclusions

- Price shouldn't be sole driver of PAS vendor decision.
- Drivers behind total cost might be different than you think. Critical:
 - Implementation timeline
 - Implementation resources
 - Long-term staffing strategies
 - **Your company's willingness to realign itself to take advantage of new capabilities**
- Policy admin is so central to business/IT strategy that more discipline than usual is required.
 - Long vendor selection projects may be necessary
 - Engage business, IT, program management, etc.

Q & A

For a copy of this presentation, contact:

Craig Weber

Celent, LLC

617-262-3129

cweber@celent.com