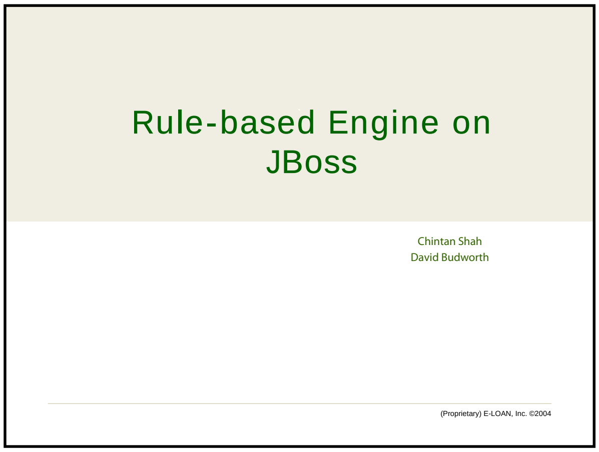
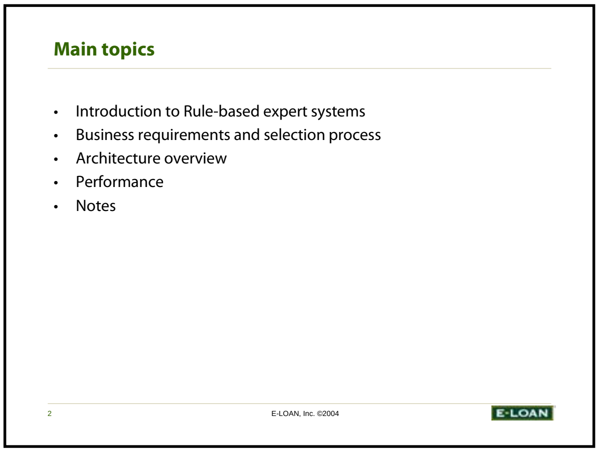


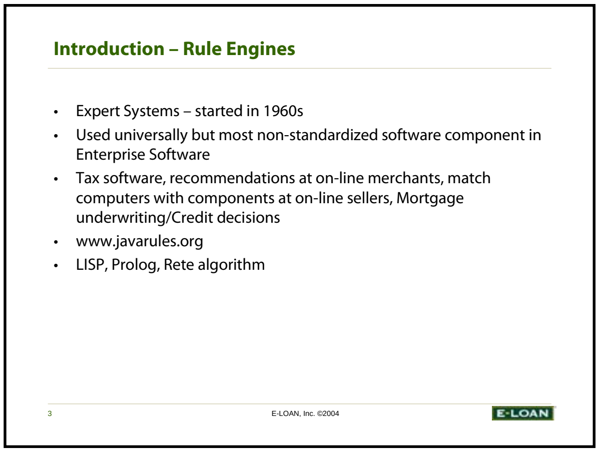
Rule-based Engine on JBoss

Chintan Shah
David Budworth



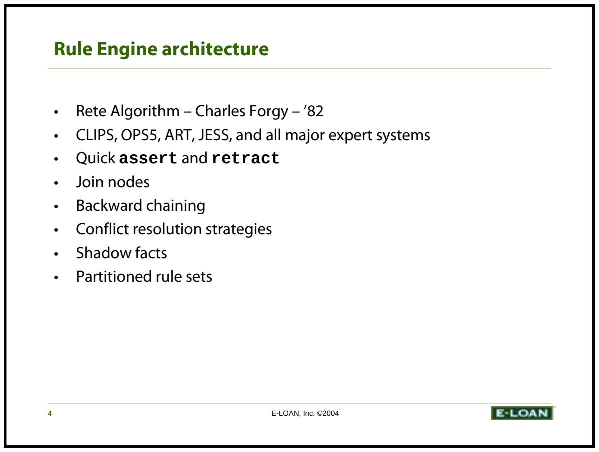
Main topics

- Introduction to Rule-based expert systems
- Business requirements and selection process
- Architecture overview
- Performance
- Notes



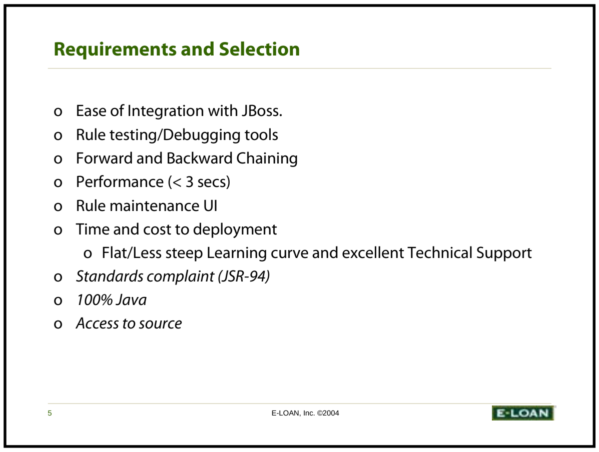
Introduction – Rule Engines

- Expert Systems – started in 1960s
- Used universally but most non-standardized software component in Enterprise Software
- Tax software, recommendations at on-line merchants, match computers with components at on-line sellers, Mortgage underwriting/Credit decisions
- www.javarules.org
- LISP, Prolog, Rete algorithm



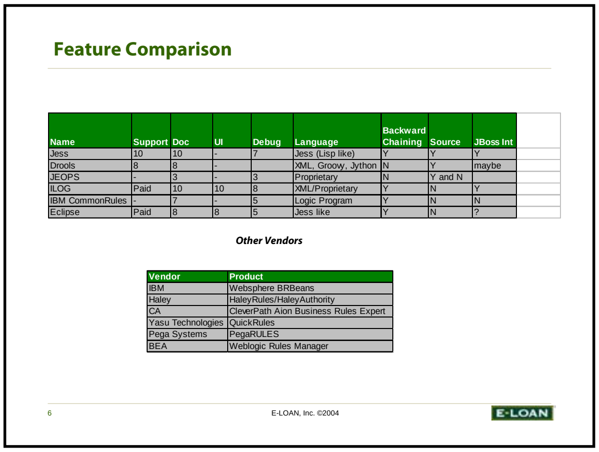
Rule Engine architecture

- Rete Algorithm – Charles Forgy – '82
- CLIPS, OPS5, ART, JESS, and all major expert systems
- Quick `assert` and `retract`
- Join nodes
- Backward chaining
- Conflict resolution strategies
- Shadow facts
- Partitioned rule sets



Requirements and Selection

- Ease of Integration with JBoss.
- Rule testing/Debugging tools
- Forward and Backward Chaining
- Performance (< 3 secs)
- Rule maintenance UI
- Time and cost to deployment
  - Flat/Less steep Learning curve and excellent Technical Support
- *Standards complaint (JSR-94)*
- *100% Java*
- *Access to source*

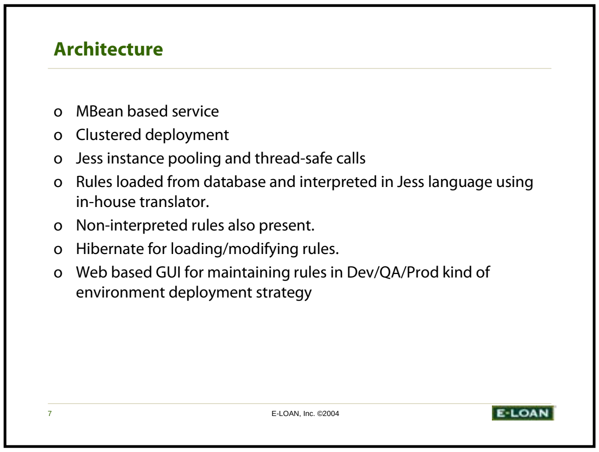


Feature Comparison

Name	Support	Doc	UI	Debug	Language	Backward Chaining	Source	JBoss Int
Jess	10	10	-	7	Jess (Lisp like)	Y	Y	Y
Drools	8	8	-		XML, Groovy, Jython	N	Y	maybe
JEOPS	-	3	-	3	Proprietary	N	Y and N	
ILOG	Paid	10	10	8	XML/Proprietary	Y	N	Y
IBM CommonRules	-	7	-	5	Logic Program	Y	N	N
Eclipse	Paid	8	8	5	Jess like	Y	N	?

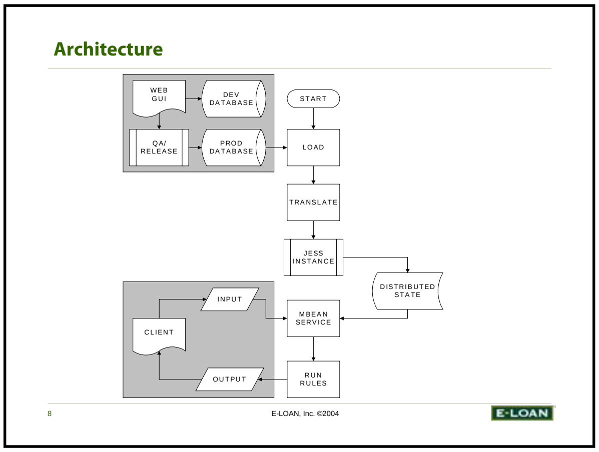
Other Vendors

Vendor	Product
IBM	Websphere BRBeans
Haley	HaleyRules/HaleyAuthority
CA	CleverPath Aion Business Rules Expert
Yasu Technologies	QuickRules
Pega Systems	PegaRULES
BEA	Weblogic Rules Manager

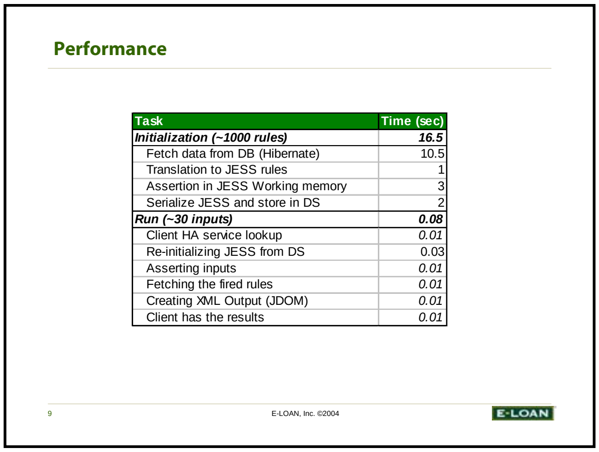


Architecture

- MBean based service
- Clustered deployment
- Jess instance pooling and thread-safe calls
- Rules loaded from database and interpreted in Jess language using in-house translator.
- Non-interpreted rules also present.
- Hibernate for loading/modifying rules.
- Web based GUI for maintaining rules in Dev/QA/Prod kind of environment deployment strategy

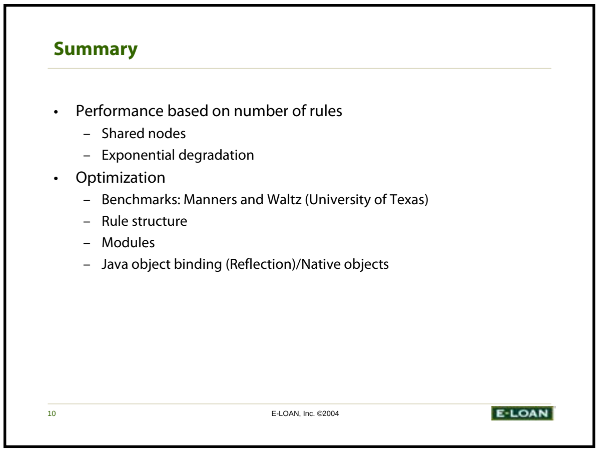


Architecture



Performance

Task	Time (sec)
Initialization (~1000 rules)	***16.5***
Fetch data from DB (Hibernate)	10.5
Translation to JESS rules	1
Assertion in JESS Working memory	3
Serialize JESS and store in DS	2
Run (~30 inputs)	***0.08***
Client HA service lookup	*0.01*
Re-initializing JESS from DS	0.03
Asserting inputs	*0.01*
Fetching the fired rules	*0.01*
Creating XML Output (JDOM)	*0.01*
Client has the results	*0.01*



Summary

- Performance based on number of rules
  - Shared nodes
  - Exponential degradation
- Optimization
  - Benchmarks: Manners and Waltz (University of Texas)
  - Rule structure
  - Modules
  - Java object binding (Reflection)/Native objects