Apex

An Adaptive Policy Engine for Next Generation Networks
Liam Fallon, Ericsson
COMPA: A Reference Architecture for Closed Loop Management
Adaptive Policy is the Concept
- Harmonize policy models
- Provide single execution environment
- Facilitate conflict processing
- Features
  - Context aware
  - Adaptive logic selection
  - Flexible clustering options
  - Flexible deployment options
  - Flexible policy deployment

Apex is the Implementation
- Fully featured policy engine
- Many logic executors
  - Java, JavaScript, Python, MVEL, Ruby
- Integrated access to context
  - shared information
- Flexible deployment options
Apex Features

- Flexibility similar to programming languages, rule engines, workflows etc.
- Fully context-aware, adaptable, and adaptive decision making
- Facilitates intelligent conflict detection and mitigation
  - at authoring time, deployment time, and runtime
- Event-sourced system
  - connectable to virtually any trigger/actioning system
- A simple, flexible, event-sourced state machine as execution model
- Light, fast, scalable policy engine
  - Engine: 131 kB, Core+services: ~800 kB (optimizable for deployment)
  - Everything (APEX + all external dependencies): ~97 MB
  - Performance: > 150k Policy Transactions per second (non-optimized system)
Deployed Policies: Policy Engine and Application

Stimulus

Policy Application

Policy Engine

Response
Deployment Details

Policy Application

Policy Engine

Application DSL(s)

Statement(s)
any abstraction

REST
HTML5

CLI
Shell

XTex
DSLs

API
APEX Model

Policy Model, including Context

JSON (also XML)

Policy Model
MEDA

Policy ECA

... Context Model

Policy MEDA

Policy ECA

... AP-EN

AP-EN Full

AP-EN Minimal

... AP-EN XYZ

Your Use Cases, Applications, or Products

Your Policy Statements

APEX Boundary

APEX Authoring (AP-AUTH) / Domain-specific tooling

APEX Policy Model
(Executable Policy Statement)
& Deployment (AP-DEP)

APEX Engine (AP-EN)
Apex Event Flow and Context

Stimulus Event (e.g. Trigger)  e^t  e^t  e^t  e^t  e^t  e^t  e^t  e^t
C^t  C^t  C^t  C^t  C^t  C^t  C^t  C^t

Response Event (e.g. Actions)  e^r  e^r  e^r  e^r  e^r  e^r  e^r  e^r
C^r  C^r  C^r  C^r  C^r  C^r  C^r  C^r

State 1
Task Selection Logic
Task
Task Logic
Parameters

State 2
Task Selection Logic
Task
Task Logic
Parameters

State n
Task Selection Logic
Task
Task Logic
Parameters

Policy

Context C

Business Goals (Real Time Change)

Domain Goals (Real Time Change)

System Context (Real Time Change)
Ericsson in ONAP
Ericsson Contribution to ONAP

- Ericsson is 7th largest contributor to ONAP
- For the Casablanca Release in Policy
  - Ericsson contributed Apex
  - Ericsson developed the Policy Deployer for all policies
- Ericsson is also contributing to the following ONAP projects
  - Integration
  - SO (Service Orchestrator)
  - APPC (Application Controller)
  - SDC (Service Design and Creation)
  - DMaaP (Data Movement as a Platform)
Source Code in ONAP Gerrit
Documentation in “Read the Docs” (Link may change after release)
Live Documentation on Ericsson GitHub
Talk at ONAP Developer Forum
• PDF at the bottom of the page

Related Papers
_______ In APEX Docs
_______ Research Gate APEX Project
Thank you and
OPEN SOURCE NETWORKING DAYS