Integrated SNMP Management

With IT/Operations and Network Node Manager

Presented by: Mike Peckar

Principal, Fognet Consulting

Agenda

Integrated SNMP Management with ITO and NNM:

- What is INSM?
- Event architecture: NNM
- Event architecture: ITO
- Integrated event architecture
- Integrated event deployment
- INSM best practices

Agenda

What’s covered:

- •Event management definitions
- •INSM definition and generalities
- •OpenView event/message arch.
- •Event daemon interactions
- •Internal/external event flows
- •Administration of INSM events
- •INSM features and failings

What’s not covered:

- •SNMP internals and politics
- •NNM/ITO application integration
- •ITO message filtering internals
- •NNM map maintenance for ITO
- •SNMP APIs/developer issues
- •Distribution or scalability issues
- •HP and 3rd party integration

What is INSM?

INSM-related terms:

- Integrated network and systems management (INSM)
- Event management
- Fault management
- Problem management
- Performance management
- Application management

What is INSM?

Historical development:

- Historically separate, today mostly still so.
- Distributed computing drove demand for INSM.
- Network management: SNMP.
- Systems mgmt: client-server-based.
- INSM first step towards higher levels of management.

What is INSM?

Network management:	Systems management:
•Emphasis on object mapping	•Events play central role
•Topology-based status	•Message-based status
•Events not guaranteed	•Guaranteed messages
•Simple, powerful agents	•Intelligent, flexible agents
•Pulls versus pushes	•Pushes versus pulls
•Unsolicited and asynchronous	•Solicited and synchronous

What is INSM?

Products:

Network Node Manager:	SNMP management tool
OpenView Windows:	Management Platform API’s
Operations Center:	Distributed systems mgmt tool No integration with NNM
IT/Operations:	INSM Integration points: • SNMP events into ITO • NNM apps assign to ITO user • Highlight in IPMAP

Event architecture: NNM

Network Node Manager and events:

- *xnmevents* is NNM foreground GUI.
- Simple: lines are read from ASCII file.
- Simple categories; simple acknowledgement.
- Customized by severity and source using *xnmtrap.*
- Actions launched by *ovactiond* at mgmt server.
- Icon status result of only node up/down by default.

Event architecture: NNM

Network Node Manager 6+ and events:

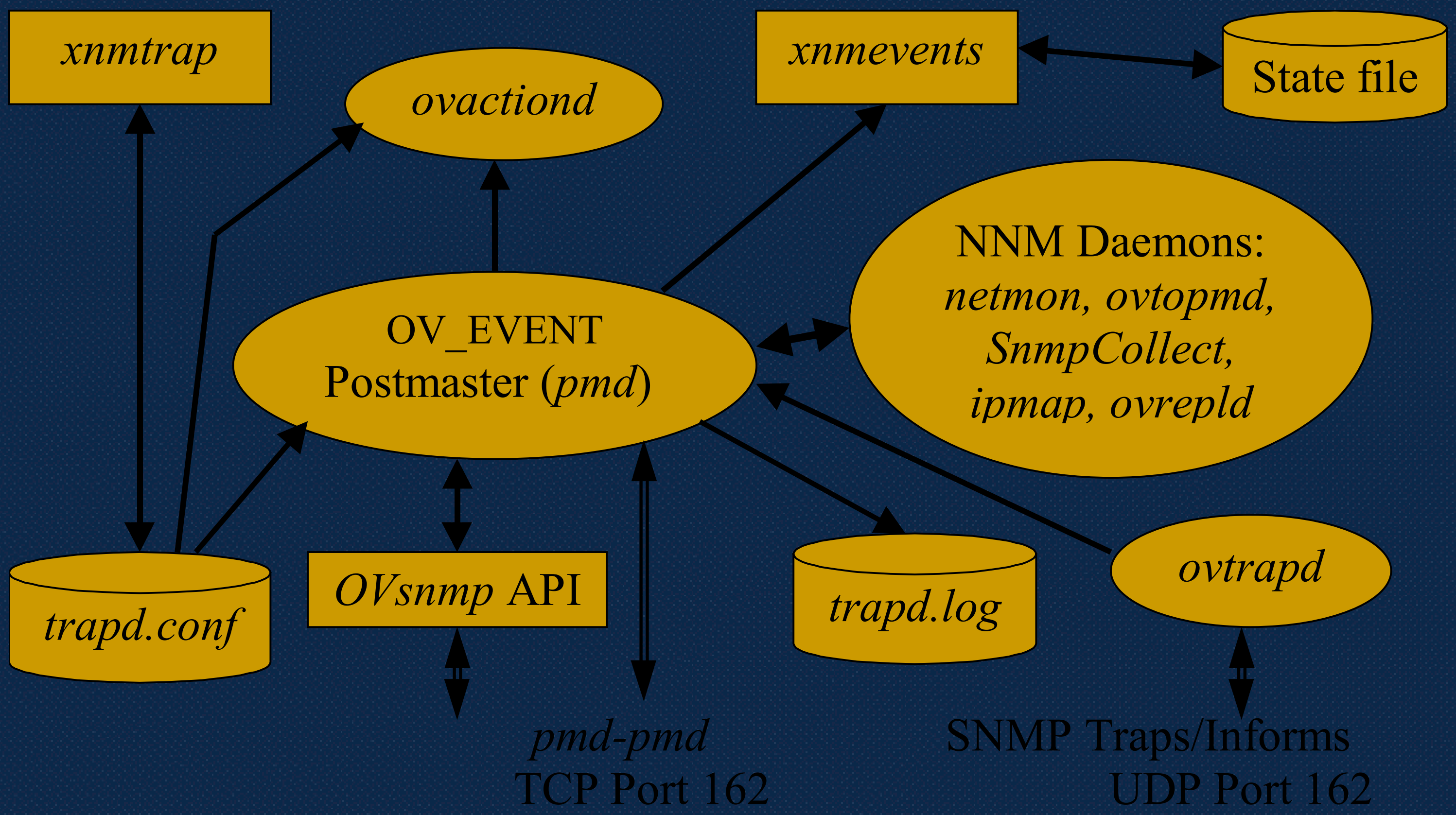
- *netmon*/ECS root cause analysis.
- *xnmevents* restructured to display events relationally.
- Data warehousing of events in new embedded DB.
- Events can be fed into same Oracle instance, but not into ITO message tablespace - need good DBA skills.
- *netmon* and *trapd.conf* backward compatible *to NNM 5*

Event architecture: NNM

NNM event-related processes and files:

- *pmd:* postmaster daemon. Receives and log events, forwards events to subscribing applications. OV_EVENT is the operative *pmd* stack.
- *trapd.conf:* defines trap formats (see man page ov_event).
- *ovtrapd:* NNM trap receptor daemon. Listens on UDP port 162 and buffers (if necessary) before sending to *pmd*.
- *xnmevents*: NNM foreground process for the event browser.

Event architecture: NNM



NNM Events Architecture

Event architecture: ITO

IT/Operations event presentation:

- User GUI’s based on matrix of Message Groups and Node Groups.
- Messages logged to RDBMS (Oracle). Active and history msgs.
- Robust message filtering via message source templates.
- Centrally administered and distributed templates and actions.
- Icons show most critical message status.
- Message ownership, links to notification systems & trouble ticket systems, and message stream API.

Event architecture: ITO

ITO messaging architecture:

- ITO agent local processing: filters, actions, logging.
- Guaranteed delivery: buffering in queues.
- Server distributes actions to other nodes.
- Actions execute as any user on any ITO agent.
- Multiple API hooks (e.g. agent or server correlation).
- Templates maintained and distributed from server.
- Robustly featured event management interface.

Event architecture: ITO

ITO messaging-related daemons:

ITO management server: *OpC* (*ovw* lrf-registered object)

opcctlm - Control manager *opcactm* - Action manager

opcmsgsm - Message manager *opcsm* - Session manager

opcdistm - Distribution manager *opcecm* - ECS manager

opcttnsm - Trouble ticket and notification manager

opcforwm - Manager to manager forwarder

ITO open agent manager: *ovoacomm* (ovw lrf-registered object)

opcmsgr - Message receiver *ovoareqhdlr* - Request handler

ovoareqsdr - Request Sender *opcmsgrd* - DCE msg receiver

Event architecture: ITO

ITO messaging-related daemons:

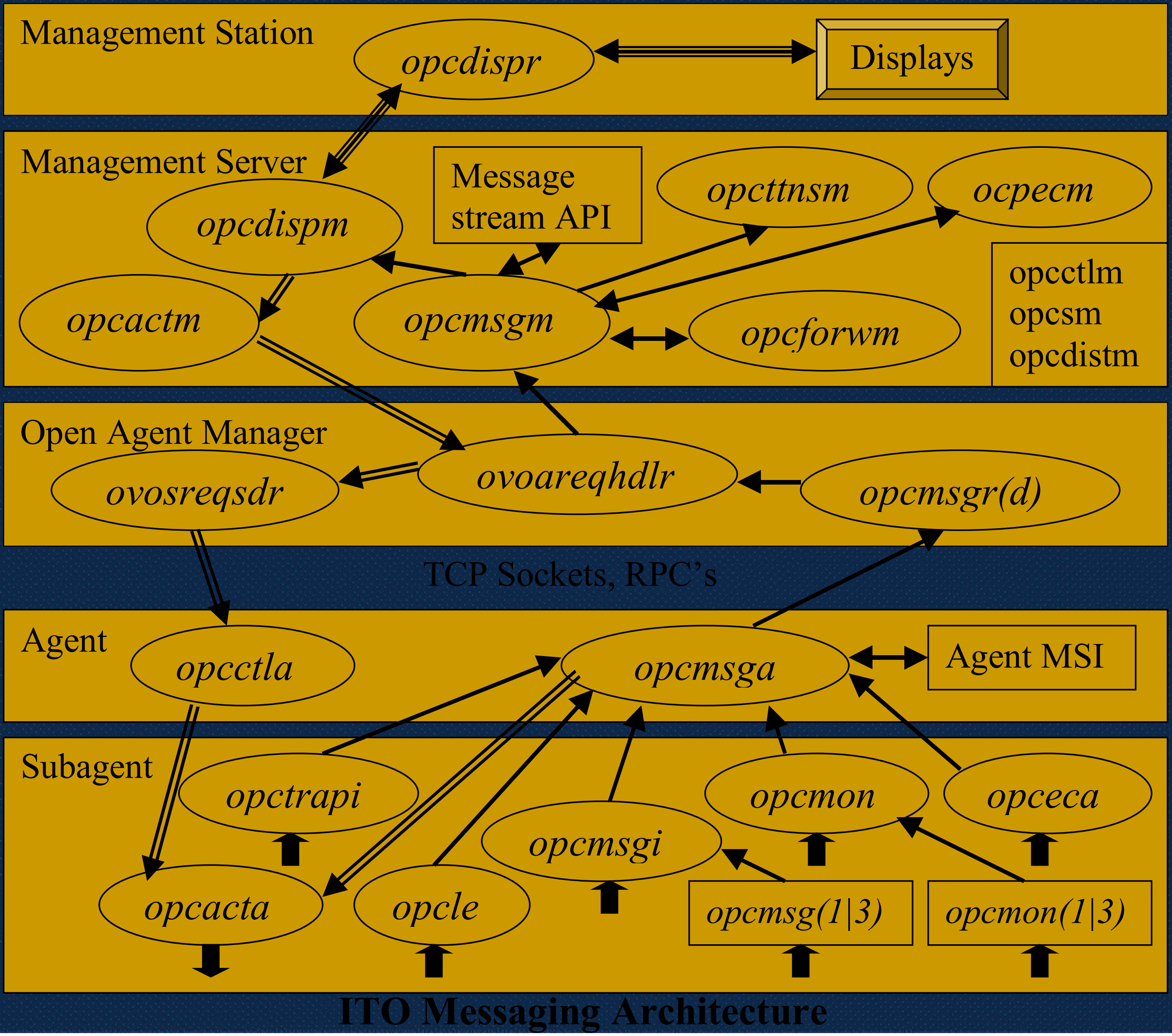
ITO agent:

opcctla	- Control agent
opcmsga	- Message agent4

ITO sub-agent:

opcacta	- Action agent	*opcsmgi*	- Message interceptor
opcle	- Logfile encapsulator	*opceca*	- ECS agent
opcmona	- Monitor agent	*opctrapi*	- Trap interceptor

Event architecture: ITO



ITO Messaging Architecture

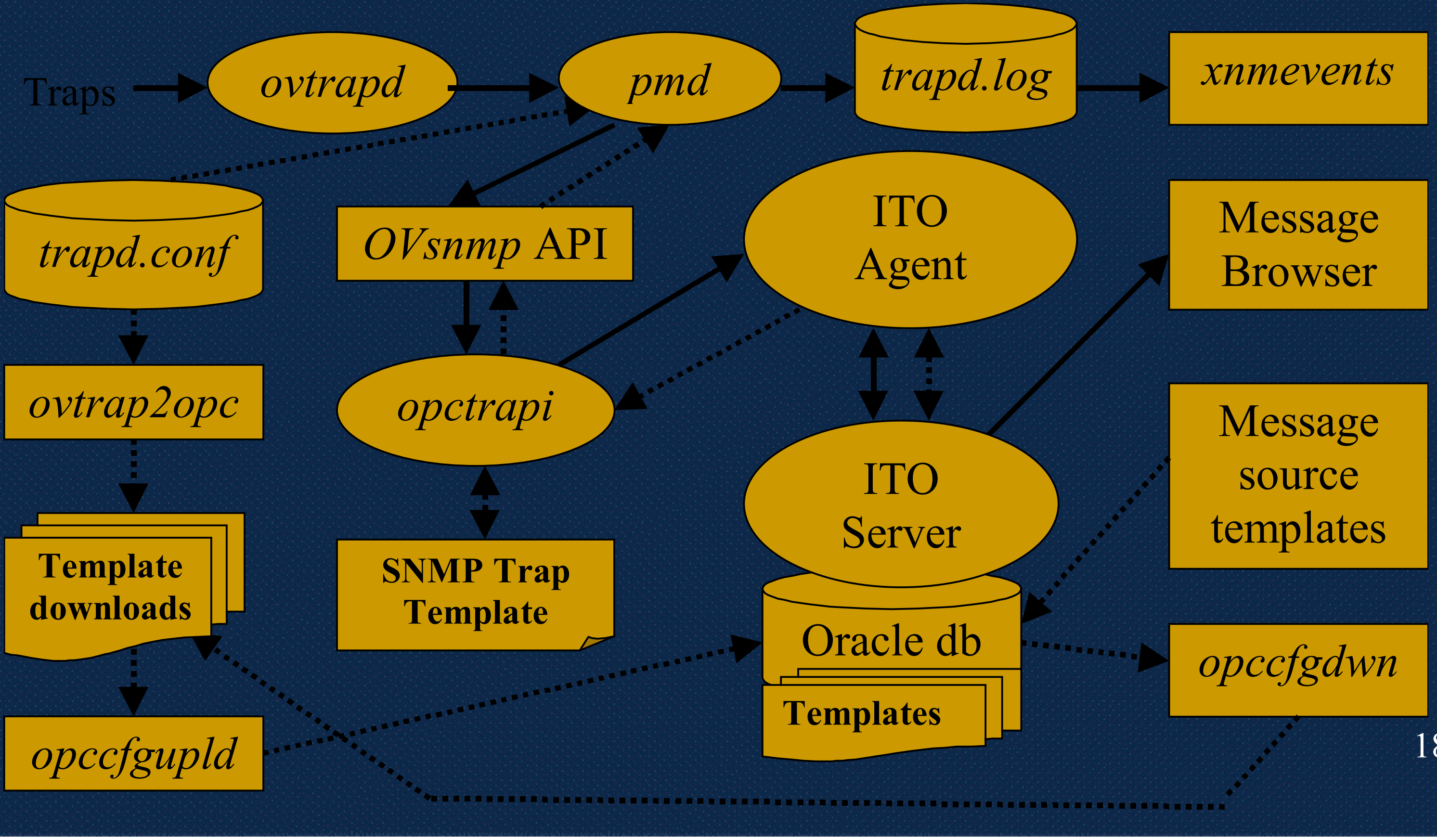
Integrated event architecture

Integrated SNMP message flow:

- SNMP messages still logged to *trapd.log*, but *xnmevents* GUI suppressed.
- *opctrapi* registers with *pmd* to receive traps based on SNMP trap template.
- ITO trap template reflects default *trapd.conf.*
- *ovtrap2opc* available for configuration updates.

Integrated event architecture

INSM event flow in NNM and ITO:



Integrated event deployment

ITO set-up to integrate SNMP traps:

- No SNMP messages by default. Trap template unassigned to management server's agent.
- Default set of node sources is Node Bank. Additional sources can be added to Node Bank or administrator can "*add node for external events*" and use IP address or wildcards: <*>.<*>.<*>.<*>
- Default Node Group is *net_devices.*

Integrated event deployment

Three strategies for INSM with SNMP:

Default SNMP trap handling scheme:

- Node scope = Node Bank, mgmt roles still separate, no central repository, no single event management interface, limited INSM.

All SNMP traps into ITO for all nodes:

- INSM. Easy set-up, message storm issue, template maintenance issue.

Best Practice: Some SNMP traps into ITO for important nodes:

- Critical traps to ITO, delete unwanted traps.
- Use *xnmevents* for tunnel-down and troubleshooting.
- Use robust message handling for important SNMP events.
- Template maintenance issue still a problem.

Integrated event deployment

OpenView INSM administration issues:

- Default ITO SNMP template.
- Cross-pollination of trap updates or additions.
- Message format incompatibilities.
- Where to perform built-in event correlation?

Understanding these limitations is first step towards choosing best practice for successful INSM.

Integrated event deployment

Default ITO SNMP template:

- Issue: All Log-only traps under NNM are placed directly in history message browser under ITO. Could fill up RDBMS tables without operator knowledge. These events are unwanted anyway.
- Resolution: Backup default template and delete all log-only traps from ITO SNMP Trap Template. (suppressing conditions or using *opccfgupld* not viable options) See procedure on slide 26.

Integrated event deployment

Cross-pollination of trap updates or additions:

- Issue: Changes made to *trapd.conf* not reflected in SNMP trap template, and vice-versa. *ovtrap2opc* integration utility designed for initial configuration, not ongoing maintenance of SNMP trap template.
- Resolution: Use procedure on slide 28 for update of multiple new trap definitions to upload to ITO, otherwise, update trap template manually.

Integrated event deployment

Message format incompatibilities:

- Issue: Trap forwarding to remote managers can only be done in *trapd.conf*. No files as node sources in ITO trap template. *ovtrap2opc* translates many event customizations incorrectly.
- Resolution: Maintain SNMP trap template manually. Maintain *trapd.conf* as well for trap forwarding and multiple node source event customizations.

Integrated event deployment

Where to perform built-in event correlation:

- Issue: SNMP traps can be correlated by NNM’s built-in ECS, ITO agent’s built-in ECS, and/or ITO server’s built-in ECS.
- Resolution: Before ITO 5, correlate closest to source with NNM ECS runtime. After ITO 5, use central ECS Designer 3 on ITO server; it operates at all three levels.

INSM best practices

Procedure for trap template customization in ITO:

- Copy SNMP trap template in mgmt server template group.
- Delete original trap template from management server template group.
- Delete unwanted *log-only* conditions; log wanted *log-only.*
- Modify template; suppress unmatched conditions.
- Add any customization in ITO trap template (manually migrate *trapd.conf* customizations).

INSM best practices

Procedure for trap template customization (cont’d):

Example Log-Only events to assign a severity:

Authentication_fail (suppress identical)

Interface_Down (for selected nodes)

Node_Up

Interface_Up

Example non-Log-Only events to delete:

OV_Station_Critical

OV_Station_ Marginal

OV_Station_ Warning

OV_Station_ Major

OV_Network_Major

OV_Network_Critical

OV_Segment_Major

OV_Segment_Critical

INSM best practices

Procedure for trap template updates in ITO:

Use when new trap definitions loaded from NNM GUI or via a third-party product such as CiscoWorks (e.g. via *xnmloadmib*):

- Backup *trapd.conf*; add new trapdefs; read man *trapd.conf.*
- diff -e trapd.conf.orig trapd.conf > trapd.opc or separate in editor.
- strip ed controls (a,c,d); add “VERSION 3” at top of file.
- $OV_BIN/OpC/utils/ovtrap2opc $OV_CONF/C/trapd.opc “MY SNMP 5.0 Traps” mytraps; answer “no” when asked to upload.
- opccfgupld -subentity -add mytraps.

INSM best practices

Some cool features of integrating traps in ITO:

- Multiple trap template support: Load a set of device-specific traps as a separate template and assign a separate msg group, etc. Be sure to “*suppress unmatched.*” (Example: OmniBack II.)
- MIB object monitors (see ITO Admin Guide.)
- Node down automatic acknowledgement with Node up event:

Automatic action for node_down condition (use node source: mgmt server):

```
echo <$MSG_ID> >/tmp/node<$2>
```

Automatic action for node_up condition (source mgmt Server, auto ack: yes):

```
$OpC_BIN/opcmsgack `cat /tmp/node<$2>`; rm /tmp/node<$2>
```

INSM best practices

Recap:

- INSM ♑ network management and system management in the same framework or on the same platform.
- INSM is difficult because different event architectures lead to message format incompatibilities.
- INSM requires organizational integration first.
- Best overall INSM solutions today integrate best fault, problem, and performance management tools with