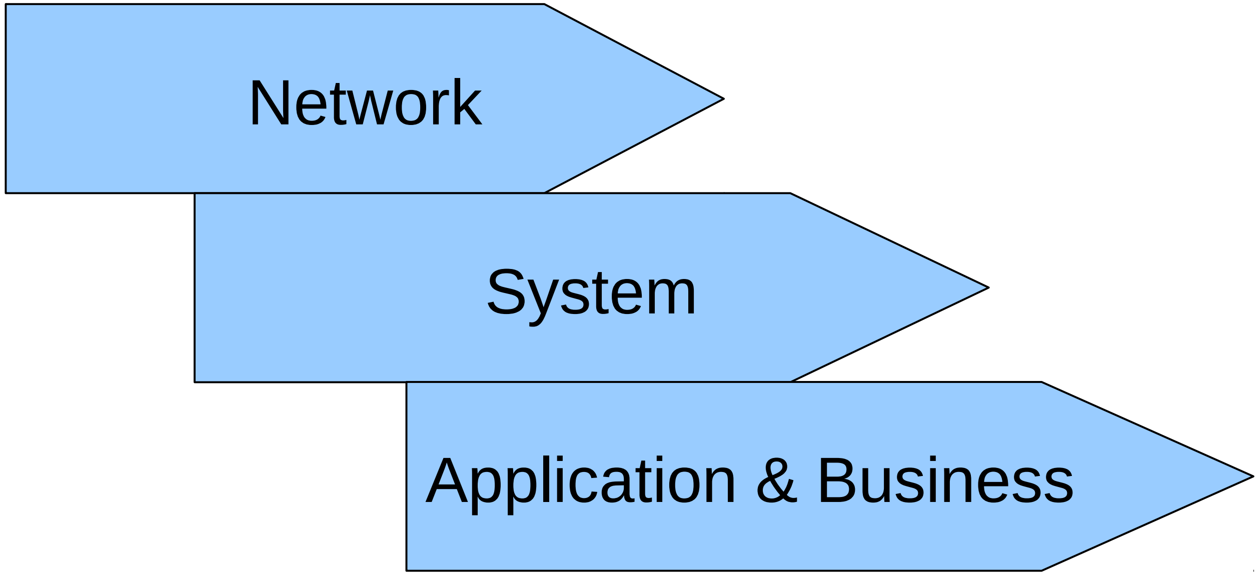
Business application monitoring - BAM

The goals of business activity monitoring are to provide ***real time information*** *about the status and results of various business related operations, processes, and transactions.*

The main benefits of BAM are to enable an enterprise to make better ***informed business decisions****,* ***quickly address problem*** *areas, and* ***re-position organizations*** *to take full advantage of emerging opportunities.*

Wikipedia

Next step in IT surveillance



Has nothing to do with IT

Warning if the delivery of gods is lower then 80% of ready.

If the number of orders drop below my daily/weekly estimate I need to know

If the ratio between web and phone orders are higher then ...

If the number of errors in incoming EDI messages are higher then 5% of total

My route planning must be over 90% of my received orders

If the number of international shipment is above 10000 at 17:00 I need to give gateway a warning

What can BAM mean for you?

- Get real measurement on what is going on inside our applications from a business perspective
- Get out of the "IT-blindness"
- Understand application to application dependencies and effects of a process disturbance
- Define thresholds on business related events
- SLA by meaningful numbers
- Business trends in real time
- Notification to the business - not just IT
- IT resource utilization in business numbers

How can this be achieved?

- What to monitor and measure
  - Business processes of importance
  - Applications involved
  - Measurable entities
- Thresholds level - alarm
- People/groups to notify
- Monitoring infrastructure
- Connecting to applications
  - "None intrusive" probing of the application events

What is different from normal monitoring?

- Business events is dynamic to it nature
  - *"The numbers of orders are **different** depending of time of day and day in week"*
- Business events is dependent to other events
  - *"The number of invoices that should be created are **depending** on the number of received orders"*
- Capability relate to multiple events
  - *"The percentage between "web orders" and "all orders" should not be lower then 80%"*

Example 1 – Monitor the number of orders received during the day

The order management application receive order 24 hours a day during Monday to Friday.

The total aggregated number of orders are different depending on time of the day. The business expect to have a total numbers of orders of 1500 at 13:00, at 14:00 the order count should be 2300, at 15:00 it should be 3400, etc. Between every hour we interpolate that the order rate are according to a linear equation. This means that the threshold at 13:20 is (2300-1500)*20/60+1500 = 1767.

The warning alarm level should be triggered between 90% and 70% of the threshold and a critical alarm if the measured value is below 70% of the threshold.

Example 2 – Monitor the number of created invoices in relation to the number of received orders

The invoice system should invoice at lest 80% of the daily incoming orders in the same day with one hour delay.

This means that the measured value of orders with one hour delay must be used as a threshold for the number of created invoices.

Example 3 – Monitor the current number of orders and if the inflow is zero we need an alarm.

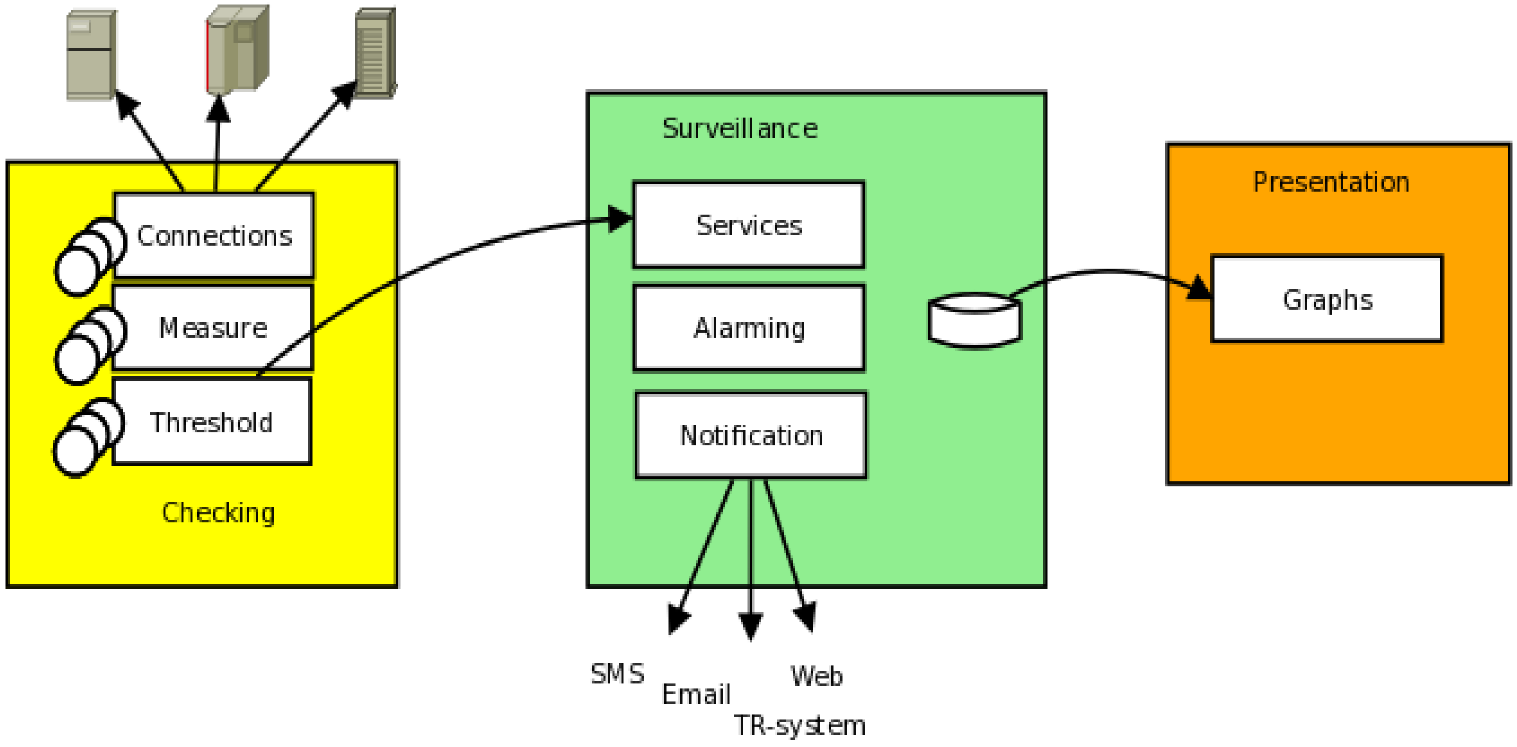
The order system have a table with all received orders, but the requirement is that we need to monitor how many that has been received during the last 10 minutes.

If this value is zero an alarm must be generated since its an indication that the sales system are not generating orders. To achieve this monitoring we use the last and the previous sample of the total number of orders from example 1 and create the difference between the two to get a new virtual entity to measure, with an threshold level of 0.

What is the same?

- Connectivity to applications and systems
- Define responsible groups
- Notification
- Escalation process
- Presentation of status and graphing
- Historical data
- .... all the features you are used to with NOC systems

BAM architecture



bischeck – a Nagios plugin

- An application surveillance and monitoring tool
- Focus on *application* and *business* related measuring and threshold management.
- Integrate with the Nagios plattform with passive checks over the Nagios integration protocol NSCA
- bischeck are license under GPL2
    - All other software used are open source license like GPL, Apache, etc
- Run as a linux service/deamon

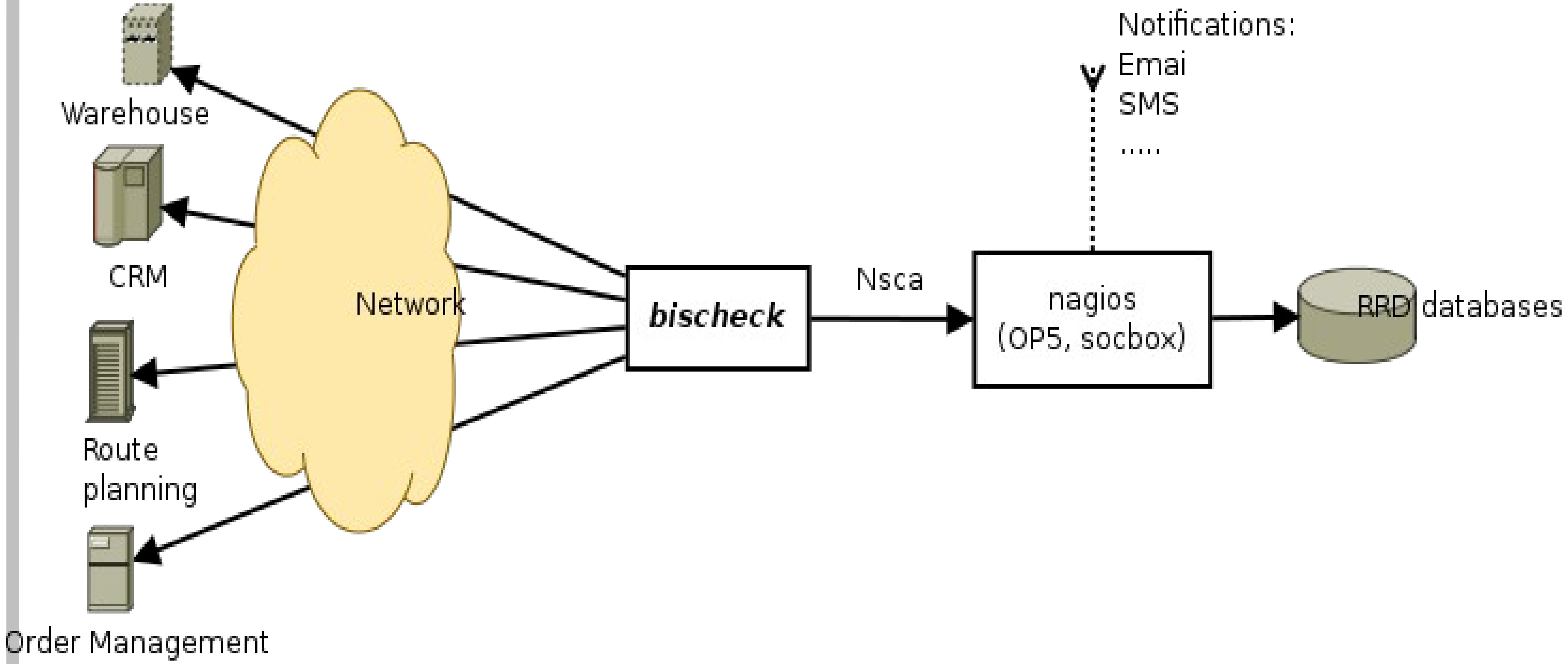
Who is using bischeck – what we know?

- DHL Freight Sweden
  - BAM for the whole forwarding and logistic process

Features

- **Different threshold** values depending of time of the day and day of month or week and holiday calender.
- **Dynamic threshold** based on measured value from other monitored services.
- **Multiple scheduling schema** per service. This enable a fine grain control of when a service should be monitored.
- Configure services that are based on multiple measured entities, what could be described as **"virtual" services**.
- Date **macros** in execution statements of measured services, typical used in a where clause when selecting from databases.
- Multitude of ways to connected to a the services to measure by allowing **custom service connection** methods.
- Support for **custom** threshold and connection classes

Overview


Warehouse
CRM
Route planning
Order Management
Network
bischeck
Nsca
nagios (OP5, socbox)
Notifications:
Emai
SMS
.....
RRD databases

Host, service and service items

- Primary configuration objects
- Host
    - Name equivalent with Nagios Host configuration
- Service
    - Define a name, same Nagios Service configuration
    - Connection method, e.g. jdbc
    - Schedule(s) - when to execute the service, cron style
- Service Item
    - Execution specification, e.g. SQL statement
    - Threshold specification, the threshold class to use.

What is a threshold

- Defines some logic that the measured value is compared against to define if an alarm should be triggered
- In business that threshold are often dynamic and/or the threshold source(s) is some other service we measure.
- Example
    - Measured value is 112 at sampling 13:45
    - Threshold configuration returns that the threshold value should be 140 at 13:45 for this service.
    - Threshold warning level is defined to be 10% of the threshold and critical level to be 30%.
    - A warning state will be set

bischeck threshold

- Threshold is the definition for a service item that alarm triggered from. 3 different configuration are available:
    - Over the threshold
    - Under the threshold
    - Interval of the threshold
- Warning/Critical is set as a percentage of the threshold
- Threshold logic is implemented as a java class and configured for each service item
    - Threshold classes can easily be custom developed

bischeck cache

- Cache for all the historical data in an LRU structure
    - Number in cache is configurable
    - host-service-servicitem[index] where index 0 is the latest measured
- Cache is used for “virtual” services and as threshold source
    - Virtual services and thresholds can be created based on mathematical expression on “real” measured services.
    - Example – to get an incremental value of a counter
        - host1-service1-servicitem1[**0**] **-** host1-service1-servicitem1[**1**]
    - Mathematical expressions is based on JEP open source version 2.3.1

Architecture - bischeck

- Java 6 based with internal JMX based surveillance
- Integrate with nagios over nsca protocol for passive checks.
- Parallel service execution through thread pool
- Run as a linux service.
- Can run on same server as nagios or a dedicated server.
- Configuration topics are host, services, service items and threshold
  - XML based configuration
  - Host and service name must be same as in nagios

Graphing

- Through pnp4nagios
- Bischeck has a pnp4nagios templates to plot measured and threshold value

System requirements

- CentOS 5.5 or RedHat 5.5 server (Linux)
  - Will be support on other platforms on community demand
- Java 6 or later

Future road map – not final

- More integration protocols then NSCA
    - E.g OpenTSDB
- Cache distribution, cache persistent
- Installation support for other linux distributions
    - Windows ?