Network Monitoring, Management and Automation

Ticketing Systems with

RT

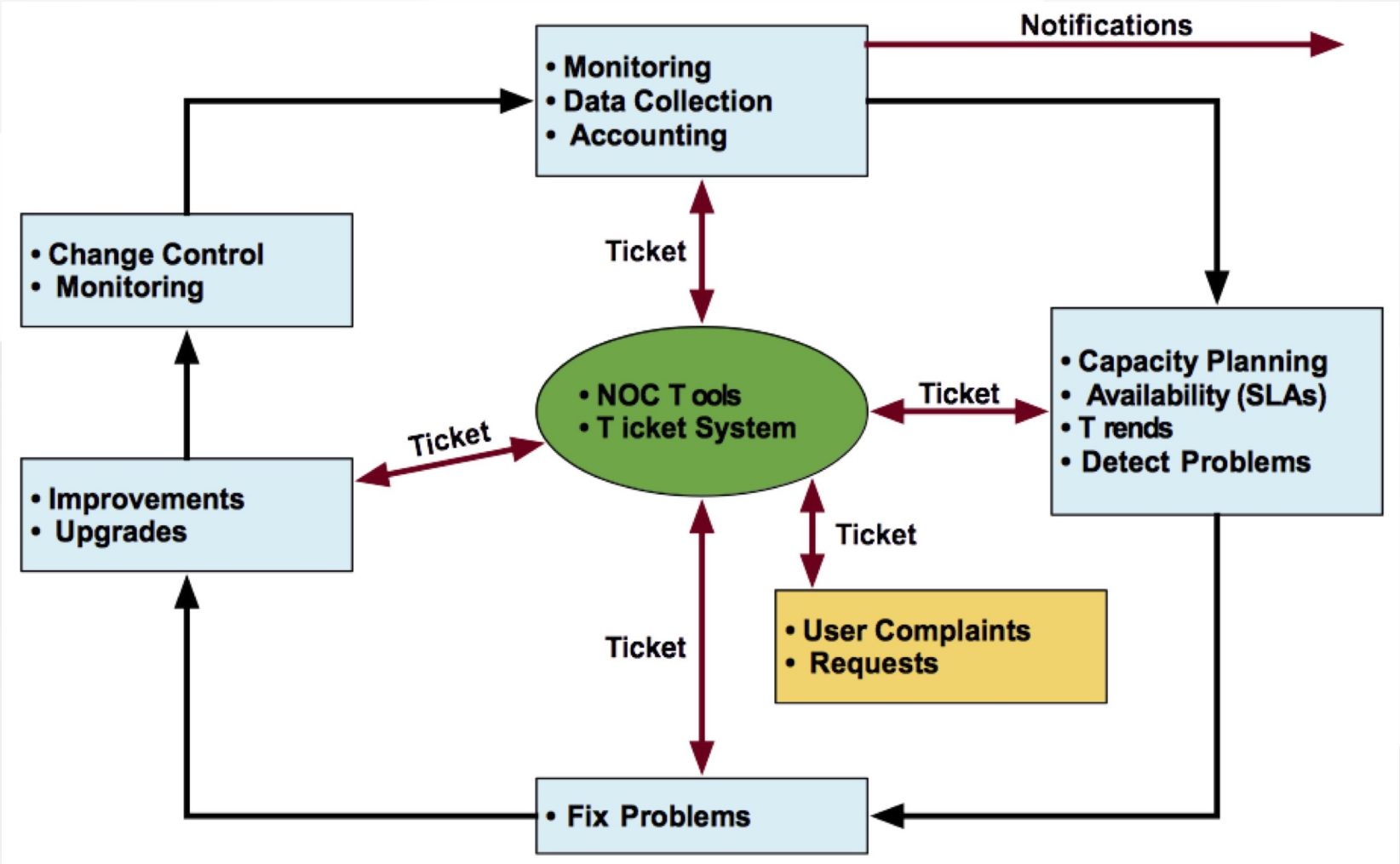
npNOG 5

Dec 8 - 12, 2019

Typical Support Scenario

- Lots of email traffic requesting help, request for services, etc
- Archived as text without classification Very difficult to find current status or problem history
- Sometimes problems were forgotten or never resolved
- Difficult for another person to follow up on a problem that someone else started dealing with

Why Ticketing Systems?



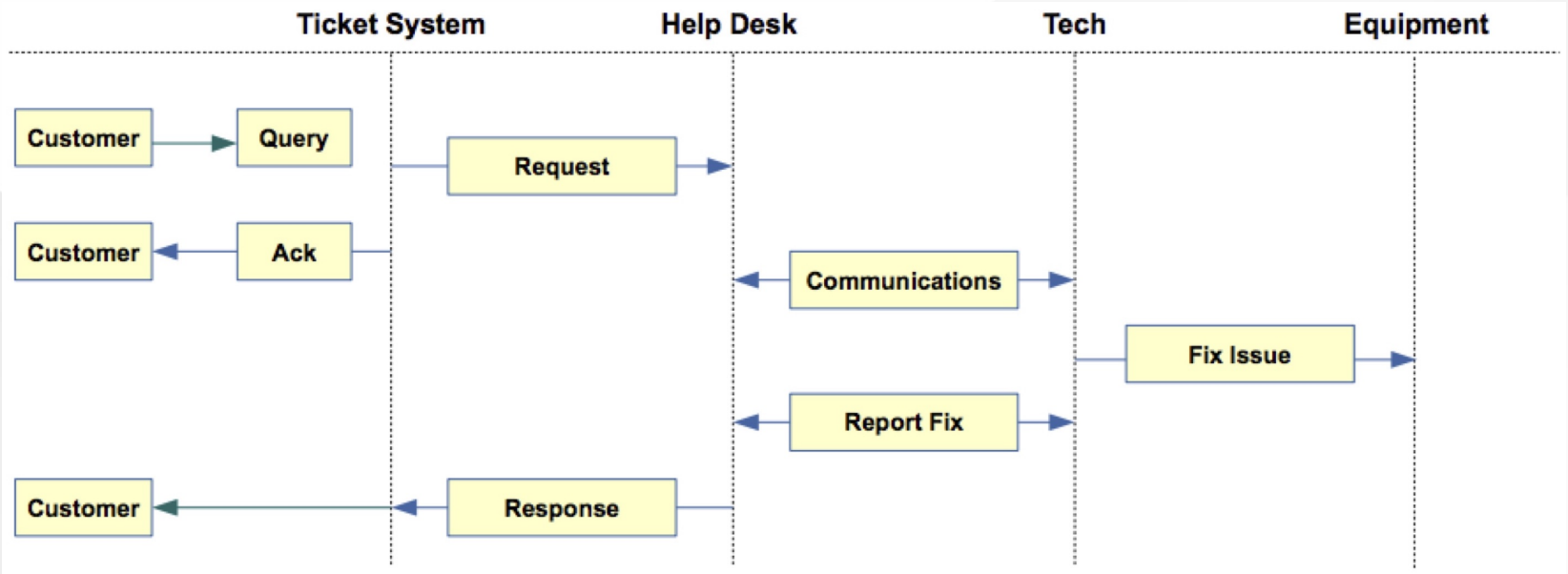
Ticketing Systems

- **Why are they important?**
  - Track all events, failures and issues
  - Focal point for help desk communication
- **Use it to track all communications**
  - Both internal and external
- **Events originating from the outside:**
  - customer complaints
- **Events originating from the inside:**
  - System outages (direct or indirect)
  - Planned maintenance, upgrades, etc.

Ticketing Systems (Contd.)

- Use a ticketing system to follow cases, including communicatoin between the support staff
- Each case is considered a ticket
- Each ticket has a ticket number
- Each ticket goes through a similar life cycle:
  - New
  - Open
  - ...
  - Resolved

Help Request with Tickets



Request Tracker (RT)

- Heavily used worldwide
- Can be customized to your location
- Somewhat difficult to install and configure
- Handles large-scale operations

RT: Features

- Several interfaces
  - Web, CLI, e-mail, etc.
- Multiuser
  - At different levels: admin, general user, guest
- Authentication and authorization
- Event history
- Notifications

RT: Advantages

- Open source and free
- Heavily used and tested
- Very active development
- Flexible
- Web interface or control via email
- Backend database (MySQL, Postgresql, Oracle, SQLite)

RT: Disadvantages

- A bit tricky to install the first time...
  - Most distributions have packages that make installation a bit easier:
    - Red Hat, Fedora, SuSE, Debian, Ubuntu, FreeBSD, etc.
- It's powerful, so you'll need to spend some time learning how it works
- Support for tracking service level agreements (SLAs) is basic

RT: Users

- Anyone who interacts with RT is a “user”
- root - Administrator with full privileges
- Privileged user (staff) - Staff who are able to operate on tickets
  - Has a password and can log in to the system
  - Less powerful than root
- Normal user (guest) - may only be able to see the status of his/her tickets
  - May or may not be able log into the system
- Nobody - default owner of new tickets

RT: Groups

- Different users have different privilege levels
- Assigning privileges to each user would be time consuming
- Easier approach: create groups of users, and assign privileges to groups
- Groups useful for other purposes as well

RT: People (Watchers, Actors)

- Each ticket has a set of people associated with it
- Requestor: who requested support
  - Usually a customer (network user)
  - But for internal tasks, requestor can be a member of the support team
- Owner: member of the support team who is responsible for the ticket at present
  - Owner of a ticket can change over its lifetime
  - Privileged users can take / assign ownership

RT: People (Watchers, Actors) (Contd.).

- cc : who gets copies of all communications between staff and requestor (responses)
  - Will see the communications, but may not be privileged to perform actions on tickets
  - e.g. : the requestors boss
- admincc: who gets copies of responses as well as internal communications between staff while working on a ticket (comments)
  - e.g. : manager of the support team

RT: Updates / Transactions

- When a ticket is being worked on, there will updates or transactions (usually via email)
- Communications between requestor and RT (staff) are called replies
- Sometimes staff need to talk internally while working on a ticket
  - These are called comments
  - Requestors don’t get copies of these

RT: Ticket States

- New: The ticket has been received by RT, but not acted upon in any way
  - RT notifies (via email) someone* of new tickets
- Open: Ticket is being acted upon
- Stalled: Progress on the ticket is stalled for some reason
  - It will hopefully come back to open state
- Resolved: Problem has been solved
  - No further action necessary

RT: Ticket States (Contd.)

- Rejected: The ticket is not our problem.
  - But records about the ticket stays in the RT database
- Deleted: The ticket does not belong on the system
  - However, records about the ticket stay in the system
- If you want to completely get rid of a ticket, you can shred it
  - Removes all database entries related to it

RT: Queues

- Queues are a way to classify the tickets
  - based on the nature of the request
  - based on the actions required
  - ....

RT: Problem Classification: Queues

- **Services**: DNS, IP addresses, Radius, LDAP
- **Security**: Attacks, scans, abuse, etc.
- **Sytems**: Email accounts, passwords, etc.
- **Networking**: Network Services Group
- **Help Desk**: Those who deal with end-users

RT: Components

- Register an event (i.e., ticket creation)
- Assign an owner
- Assign interested parties (watchers)
- Maintain change history
- Inform interested parties of each change
- Initiate activities based on status or priority

RT: Scrips (actions)

- **Create automatic actions for queues**
- scrips are “snippets of Perl code”
- Help automate things inside RT
- Take action X when condition Y occurs
  - when a staff member responds to a ticket owned by nobody, make her the owner of ticket
  - page everyone when the priority of a ticket becomes level X
- Details on how to use Scrips: http://requesttracker.wikia.com/wiki/Scrip
- See “Extensions” at the end of this presentation.

RT Configuration

- **Two Options**
  -Virtualhost: http://rt.host.fqdn
  - Subdirectory: http://host.fqdn/rt/
- **Root user ('root')**
  - Change the default password on first login ('password')
  - Assign the complete email for the root account: root@host.fqdn
  - Assign all user rights: Global -> User Rights

RT: User Creation

- Create a userid for each member of your team
- Assign privileges to each user

RT: Create Groups

Create groups of users:

- Administering privileges by group is more efficient than doing so for each user.

RT: Create Queues

Create queues for problem categories:

For example

- **Security**
- **Accounts**
- **Connectivity**

Assign users to groups and groups to each queue

- Different between AdminCC and CC
- Don't forget to create email aliases for each queue

RT: rt-mailgate

rt-mailgate facility lets us:

- Define virtual users on the RT server that correspond to ticket queues in RT.
- Allow third-party software (Nagios, Cacti, Smokeping, etc.) to automatically generate tickets in specified queues via email.
- Provide a simple interface through which end- users can communicate with your support organization via RT.
- More details at https://www.bestpractical.com/docs/rt/4.0/rt-mailgate.html

RT: Extensions

Extend the functionality of RT. For example:

- Send daily emails to remind users of tickets that have not been “taken”
- Send daily emails to each user reminding them of their pending tickets.
- Periodically increment ticket priority
- You can execute commands via email

Find extensions here:
https://metacpan.org/search?q=RT%3A%3AExtension&search_type=modules

Alternatives

- **Bugzilla**: http://www.bugzilla.org/
- **Cerberus**: http://www.cerberusweb.com/
- **Eticket**: http://www.eticketsupport.com/
- **Itracker**: http://www.itracker.org/
- **Jutda Helpdesk**: http://www.jutdahelpdesk.com/
- **Mystic**: http://www.hulihanapplications.com/projects/mystic
- **OTRS**: http://otrs.org/
- **OsTicket**: http://osticket.com/
- **Simple Ticket**: http://www.simpleticket.net/
- **Trouble Ticket Express**: http://www.troubleticketexpress.com/

References

- Best Practical Web site
  http://bestpractical.com/rt
- RT Essentials. Dave Rolsky et al. O'Reilly Media, Inc.
- Contributions to RT:
  http://requesttracker.wikia.com/wiki/Contributions
- Extensions
  http://requesttracker.wikia.com/wiki/Extensions
  http://bestpractical.com/rt/extensions.html
- Scrips
  http://requesttracker.wikia.com/wiki/Scrip
  http://requesttracker.wikia.com/wiki/ScripAction