



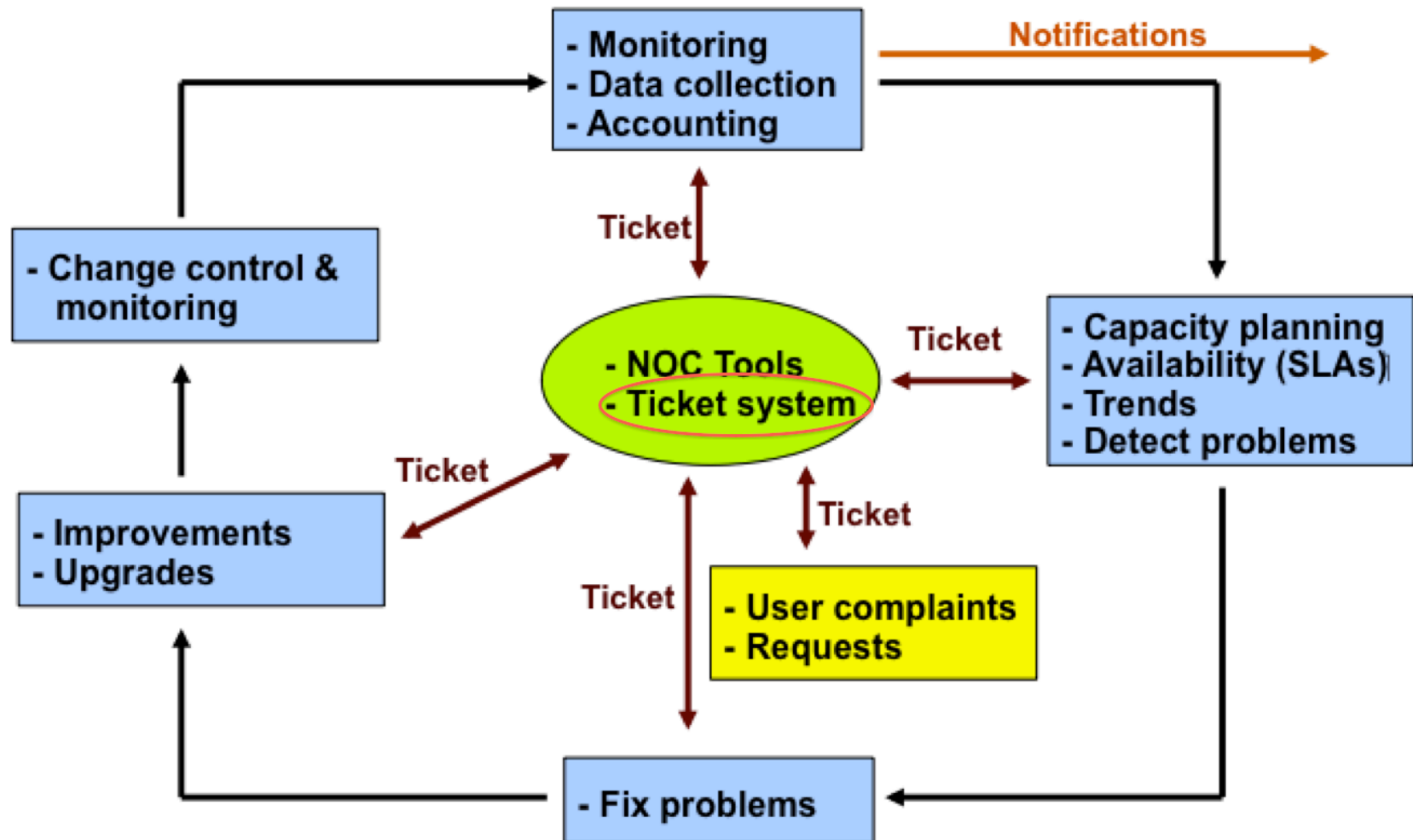
# Network Management & Monitoring

## Ticketing Systems with RT



These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license  
(<http://creativecommons.org/licenses/by-nc/3.0/>)

# Why Ticketing Systems?



# Ticketing Systems

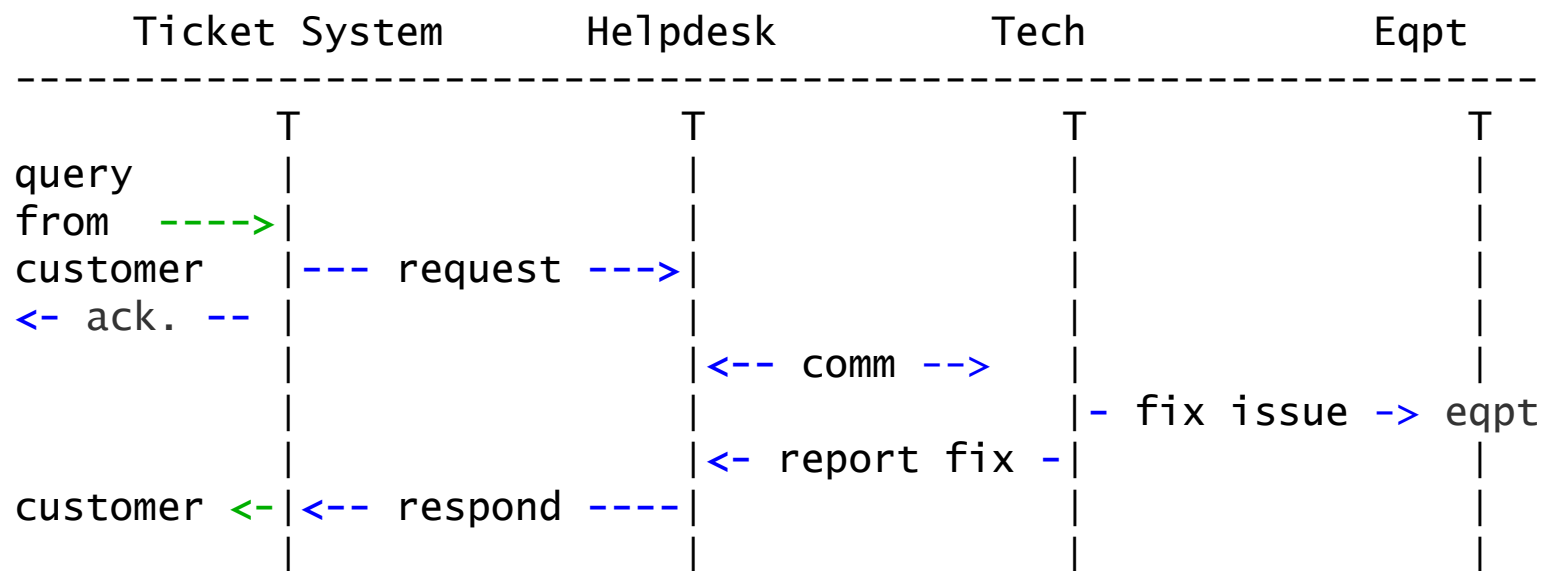
- Why are they important?
  - Track all events, failures and issues
- Focal point for help desk communication
- Use it to track all communications
  - Both internal and external
- Events originating from the outside:
  - customer complaints
- Events originating from the inside:
  - System outages (direct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

# Ticketing Systems cont.

- Use ticket system to follow each case, including internal communication between technicians
- Each case is assigned a case number
- Each case goes through a similar life cycle:
  - *New*
  - *Open*
  - ...
  - *Resolved*
  - *Closed*

# Ticketing Systems cont.

## Help Request with Tickets



# Request Tracker / Trac

## RT

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



## trac

- A hybrid system that includes a wiki and project management features.
- Ticketing system not as robust as rt, but works well for web-only ticket interface.
- Often used for "trac"king group projects.
- Used for this course:

<http://noc.ws.nsrc.org/wiki/>



# A few others...

- **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 (Open source Ticket Request System)**  
<http://otrs.org/>
- **osTicket**  
<http://osticket.com/>
- **Simple Ticket**  
<http://www.simpleticket.net/>
- **Trouble Ticket Express**  
<http://www.troubleticketexpress.com/open-source-software.html>



# RT: Request Tracker

<http://bestpractical.com/rt/>



# What's it Look Like?

The screenshot shows the RT web interface in a Mozilla Firefox browser window. The page title is "RT at a glance - Mozilla Firefox (Build 2008061004)". The browser address bar shows "RT for example.com" and the user is logged in as "root".

The main navigation menu on the left includes: Home, Simple Search, Tickets, Tools, Configuration, Preferences, and Approval.

The main content area is titled "RT at a glance" and features a "New ticket in" button, a dropdown menu set to "General", and a search box.

The dashboard contains several widgets:

- 10 highest priority tickets I own:** A table with columns #, Subject, Priority, Queue, and Status. It lists two tickets: "Office has run out of coffee" (Priority 0, General queue, pending 1 other ticket) and "order more coffee" (Priority 0, General queue, pending 1 other ticket).
- 10 newest unowned tickets:** A table with columns #, Subject, Queue, Status, and Created. It lists one ticket: "Obtain Series-C funding" (General queue, new status, created 16 min ago).
- Bookmarked Tickets:** A table with columns #, Subject, Priority, Queue, and Status. It lists one ticket: "Office has run out of coffee" (Priority 0, General queue, pending 1 other ticket).
- Quick ticket creation:** A form with fields for Subject, Queue (set to General), Owner (set to root), and Content. A "Create" button is at the bottom.
- Reminders:** A widget with an "Edit" button.
- Quick search:** A table with columns Queue, new, open, and stalled. It shows 3 new tickets in the General queue and 0 open or stalled tickets.
- Dashboards:** A table with columns Name and Subscription. It shows "SLA Performance" with a subscription of "daily at 06:00".
- Refresh:** A widget with a dropdown menu set to "Don't refresh this page." and a "Go!" button.

# Ticket Management Systems

- Why do we use the term “ticket”?
- In order to resolve a problem...
  - Who wants what?
  - Who's going to work on this?
  - When did they ask, when was it done?
  - How much time did it take (billing, hours)?
  - What's left to do?
  - Everything is summarized and presented in a simple and intuitive manner.

# Applications

- User support
- Security problem management
- Issue Tracking / Incident Management

# Essential Functionality

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

# Components

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

# 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.

# RT: Advantages

- Open source and free
- Heavily used and tested
- Very active development
- Flexible
- Web interface or control via email

# RT: Disadvantages

- A bit tricky to install the first time...
- It's powerful, so you'll need to spend some time learning how it works.
  - Most distributions have packages that make installation a bit easier:
    - Red Hat, Fedora, SuSE, Debian, Ubuntu, FreeBSD, etc.



# Problem Classification: Queues

RT allows you to create queues so that problems are classified by type:

- **Services:** DNS, IP addresses, Radius, LDAP
- **Connectivity:** Communications infrastructure problems
- **Security:** Attacks, scans, abuse, etc.
- **Systems:** Email accounts, passwords, etc
- General help

# Web Server 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](mailto:root@host.fqdn)

- Assign all user rights:

Global -> User Rights

# User Creation

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

# Create Groups

## **Create groups of users:**

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

# Create Queues

## Create queues for problem categories

- For example
  - security
  - accounts
  - connectivity
- Assign users to each queue
  - Different between AdminCC and CC
- Don't forget to create email *aliases* for each queue

# rt-mailgate

A critical component of RT. The 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.

# Scripts (actions)

## For each queue create automatic actions

- There is a group of scripts that apply to all queues.
  - Possible to customize per queue or globally
  - “*scripts*” are “snippets of Perl code”

# Extensions

You can 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

<http://wiki.bestpractical.com/index.cgi?Extensions>



# 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>

