
Ticketing Systems with RT



<https://bestpractical.com>

Ticketing Systems

What is a ticketing system?

Simply a program (software) that a **customer support team** uses to create, manage, and maintain a list (or lists) of customer problems

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

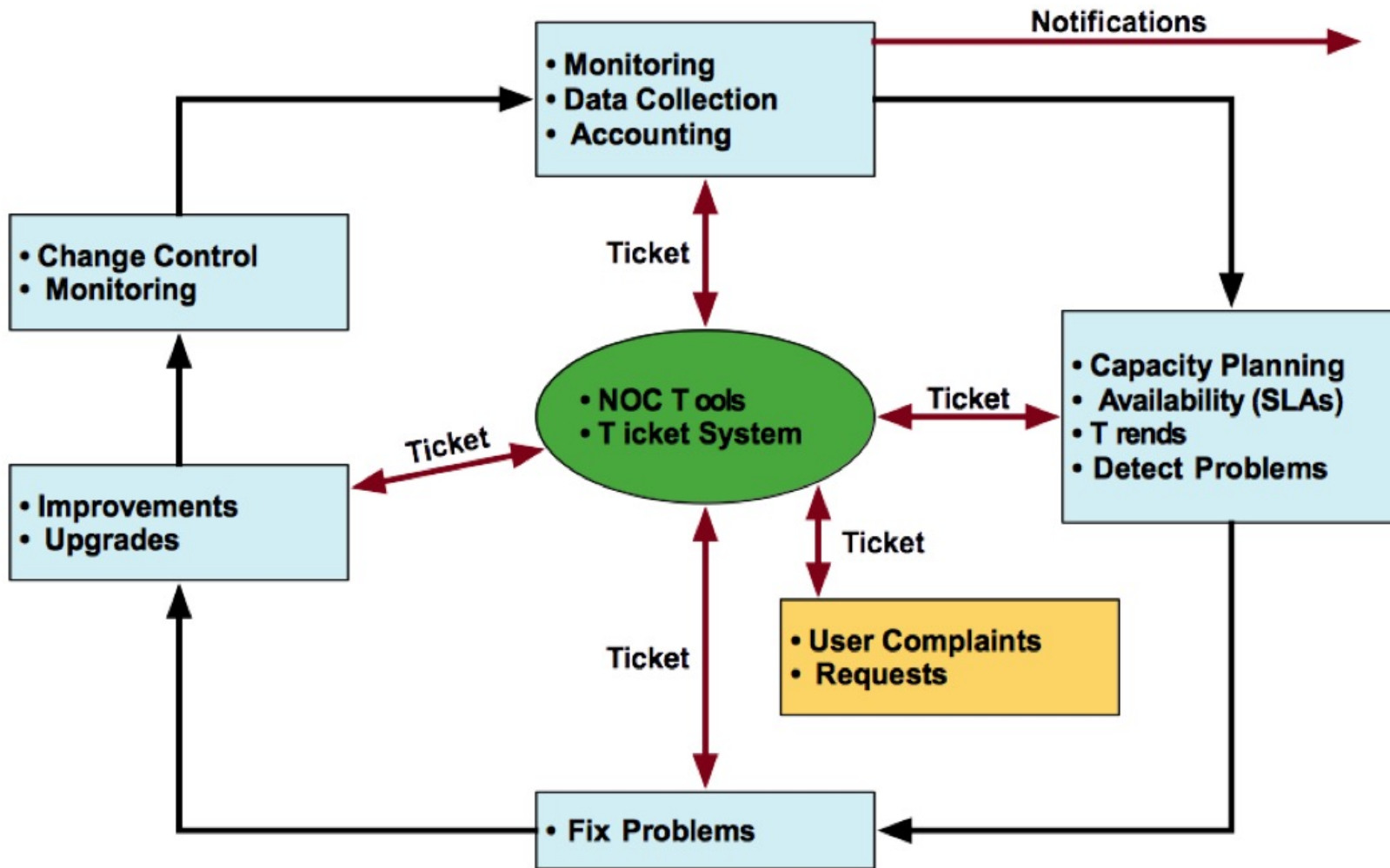
Events originating from the inside:

- System outages (direct or indirect) Planned maintenance, upgrades, etc.

Where we use Ticketing Systems

- Network operations
- Workflow
- Work orders
- Helpdesk
- Call centre
- Bug tracking

.....And many more....



Ref:<https://nsrc.org/workshops/2019/ubuntu-net-nren-noc/netmgmt/en/ticketing/ticketing-rt.pdf>

Ticketing Systems(contd)

- Each case is considered a ticket
- Everything has a unique ID
- Everything has a timestamp
- History can't be edited or deleted
- Each ticket goes through a similar life cycle
New > Open....>... Resolved

What – RT/TRac

Request Tracker

- License : GNU GPLv2
- Continuous Development since 1996
- Use for incident response
- Issue tracking and workflow platform
- Can be customized to your location
- Handles large-scale operations



RT - Advantages

- Open source and free
 - Also commercial version is available with best practical support
- Heavily used and tested
- Very active development
- Flexible
- Web interface or control via email
- Backend database (MySQL, Postgresql, Oracle, SQLite)

RT - Disadvantages

- Take time to install
 - Ex : Installing missing dependencies
 - Configuring system with many failures
- Take time to get familiar on the Ticketing system

RT - Functions

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

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
 - Able to log in to the system with passwords
 - Less privileges than admin user
- Normal user (guest) – may only be able to see the status of the own tickets
- Nobody – default *owner* of new tickets

RT : Groups

- Different users have different privilege levels
 - root
 - user(staff)
- User defined
- Used for ACLs and sharing
- Easier approach: create groups of users, and assign privileges to groups

RT : Roles

Per Ticket

- Requestor
- Cc
- AdminCc
- Owner

Per Queue

- Cc
- AdminCc

RT - People

- 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, managers
- 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 : ACLs

- User Based
- Group –Based
- Role Based

RT : Updates

- Updates or transactions are tracked(via emails)
- Communications between requestor and RT (staff) are called *replies*

RT – Ticket States

- New: The ticket has been received by RT, but not acted
- Open: Ticket is being acted upon
- Stalled: Progress on the ticket is stalled
- Resolved: Problem has been solved No further action necessary
- 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

Note : 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

- Basic Grouping of tickets
- Separate business logic
- Separate ACLs
- Separate notifications

Problem Classification

Services : DNS, IP addresses, Radius, LDAP

Security : attacks, Scans, abuse, etc

System : email accounts, Passwords, etc

Networking: Network services Group

Help Desk: Those who deal with end users

RT - Components

- Ticket creation
- Assign an owner for the ticket
- Assign interested people(Managers, Support team)
- Maintain change history
- Initiate activities on status or priorities

RT - Mailgate

- 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 the support organization via RT

More details: <https://docs.bestpractical.com/rt/5.0.1/rt-mailgate.html>

RT - Requirements

- Perl
- Database:
 - MySQL
 - PostgreSQL
 - Oracle
 - SQLite(for testing)
- Webserver
 - We can install nginx, apache2, But....
 - FastCGI
 - mod_perl
 - either have to be enabled to handle the requests

Let's try to install RT in a virtual machine which is Ubuntu installed

RT - References

Best Practical Website:

<https://rt-wiki.bestpractical.com/wiki/ManualInstallation>

NSRC :

<https://nsrc.org/workshops/2019/mnnog1/nmm/netmgmt/en/ticketing/ticketing-rt.pdf>

Itnixpro website :

<https://itnixpro.com/install-request-tracker-rt-on-ubuntu/>

Slideplayer :

<https://slideplayer.com/slide/11276505/>