

# Developing a Ticket Management SaaS solution

**Marjan Gusev, Ss. Cyril and Methodius University, 1000 Skopje, Macedonia**  
**Sasko Ristov, Ss. Cyril and Methodius University, 1000 Skopje, Macedonia**  
**Pano Gushev, Innovation, 1000 Skopje, Macedonia**

# Introduction

- Some software development companies cannot afford expensive systems that will improve processes in company or relationships with customers
- Customer relation management (CRM) software solutions usually use an approach where a software solution keeps track and manages with issues that happen between the company and customers

# Issue Tracking System

- Central role in many software projects
- Allows users to communicate with developers to let them know about any problem and to request new features
- Developers can keep track of any unresolved issues or bugs, and request more information from users
- Two types of solutions: Self-hosted and Cloud



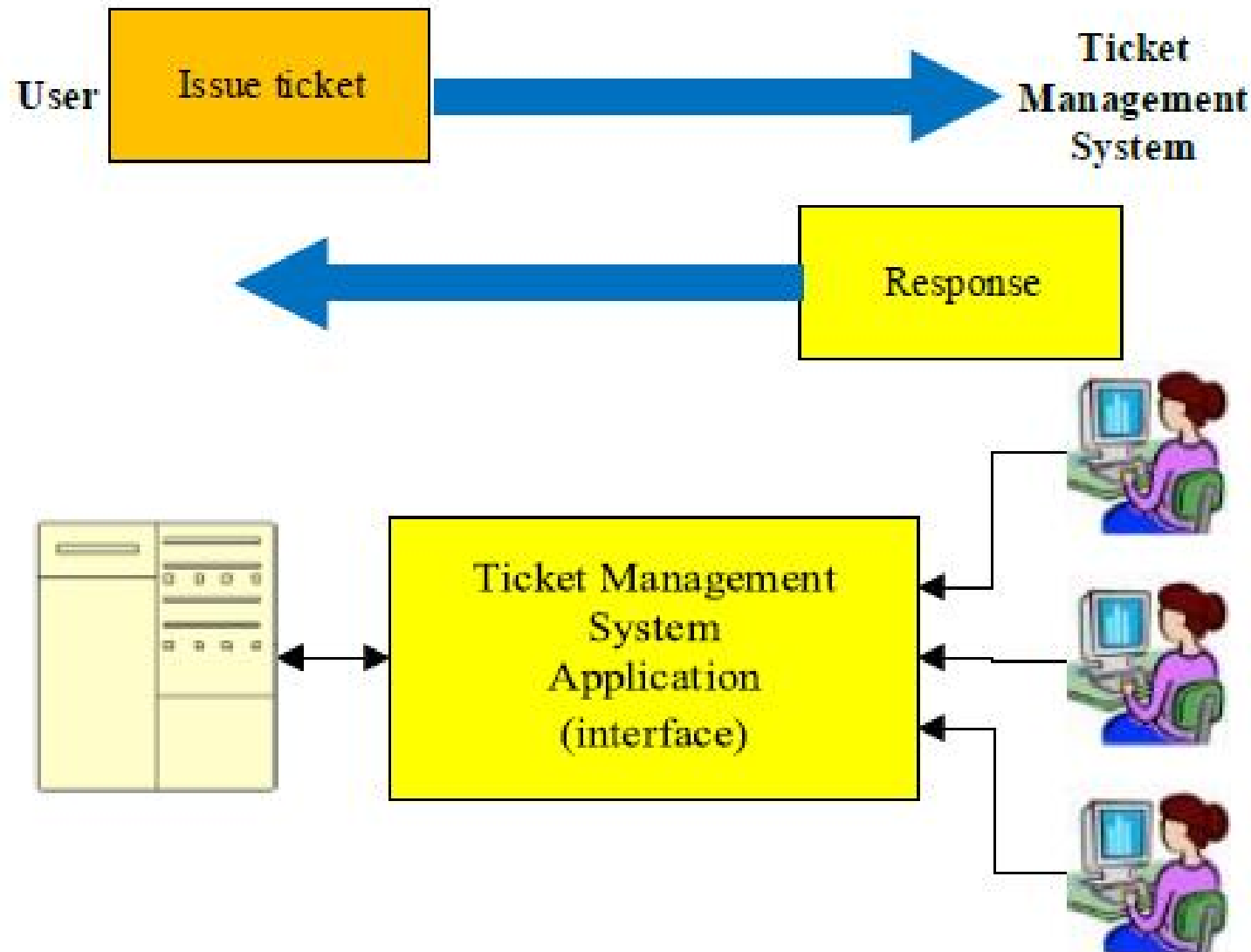
# Self-hosted solutions

- Commonly free solutions
- Non IT companies still have the problem of installation, management, maintenance and upgrades of these applications
- Require appropriate hardware and software architecture, which is an extra investment for non IT companies

# Cloud solutions

- Solve the problem for both IT and non IT companies
- Reduce costs
- Maintenance and upgrades, do not worry the users
- No start investments needed
- Companies needs only internet and browser to use them

# Design of cloud a Ticket Management System



# TMS User groups and segments

- User groups
  - Application administrator (AM)
  - Manager (PM)
  - Resolver (RE)
  - Customer manager (CM)
  - Customer (CU)
- Segments
  - Main application management (AM)
  - Provider (PM, RE)
  - Customer (CM, CU)

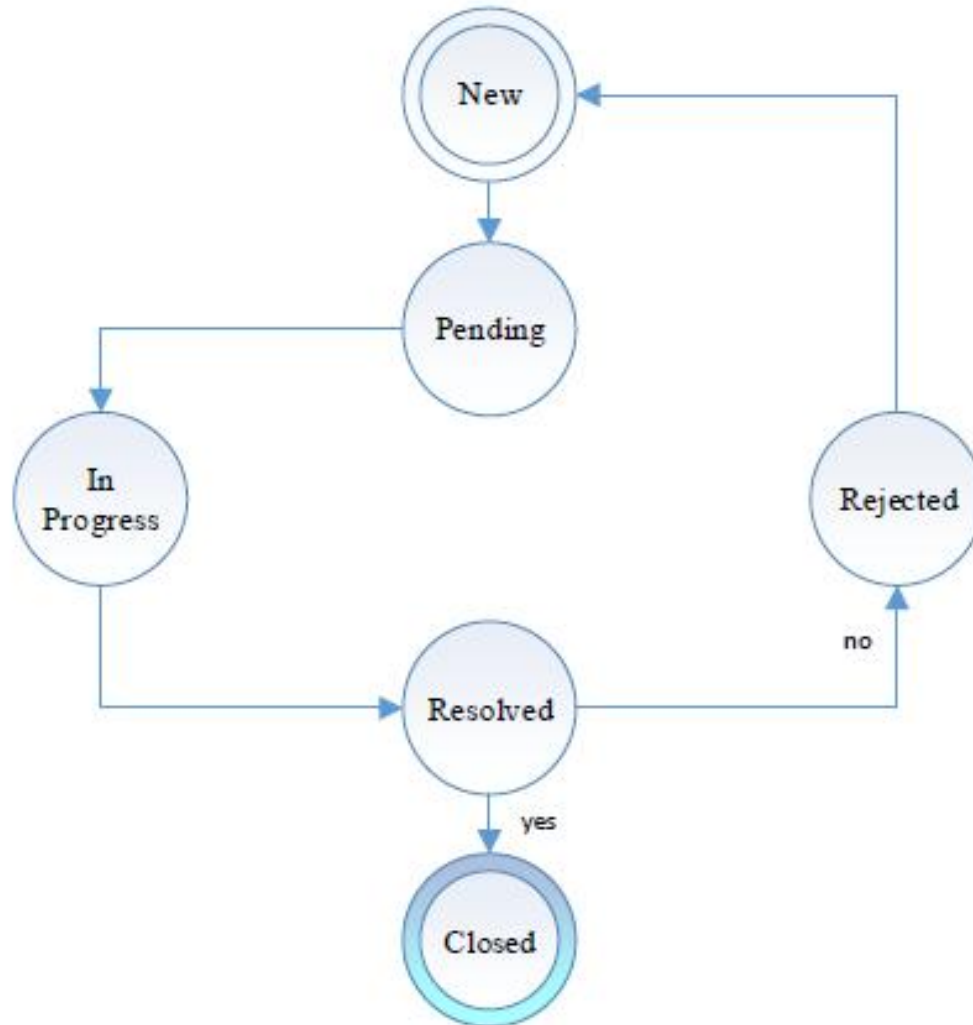


# User Access Rights

ID	Functions	AM	PM	RE	CM	CU
F1	Registration of company	Yes	No	No	No	No
F2	Customer administration	Yes	No	No	Yes	No
F3	Bug reporting	No	No	No	Yes	Yes
F4	Bug status administration	No	Yes	Yes	No	No
F5	Approving of corrections	No	No	No	Yes	Yes
F7	Create test cases	Yes	Yes	Yes	Yes	Yes
F8	Execute test cases	Yes	Yes	Yes	Yes	Yes
F9	Summary reports	No	Yes	Yes	Yes	Yes
F10	Detailed reports	No	Yes	No	Yes	No
F11	Site analytics	Yes	Yes	No	Yes	No
F12	Email notifications	No	Yes	Yes	Yes	Yes



# Ticket life-cycle



# Functional description

- Registration of a company
- Customer administration
- Administration of resolvers
- Ticket issuing
- Resolving

# ANS functionalities

- Registration of a company
- Customer administration
- Administration of resolvers
- Ticket issuing
- Resolving
- Approving
- Creating test cases
- Summary reports
- Detailed reports
- Site analytics
- Email notification

# Functional description

- **Registration of a company** - AM fills all necessary details about the company and sends them to the database. TMS saves those details in the database and returns a message to AM.
- **Customer administration** - A user sends request to TMS with user details. TMS receives those details and updates the database. Then it returns a response message.



# Functional description

- **Administration of resolvers** - A user sends a request to TMS with user details. TMS receives those details and updates the database. Then it returns a response message.
- **Ticket issuing** - CU sends a request to TMS with details. TMS receives those details and updates the database. Then it returns a response message.
- **Resolving** - RE responds to the ticket. TMS updates the database and returns the response message to CU.

# Functional description (2)

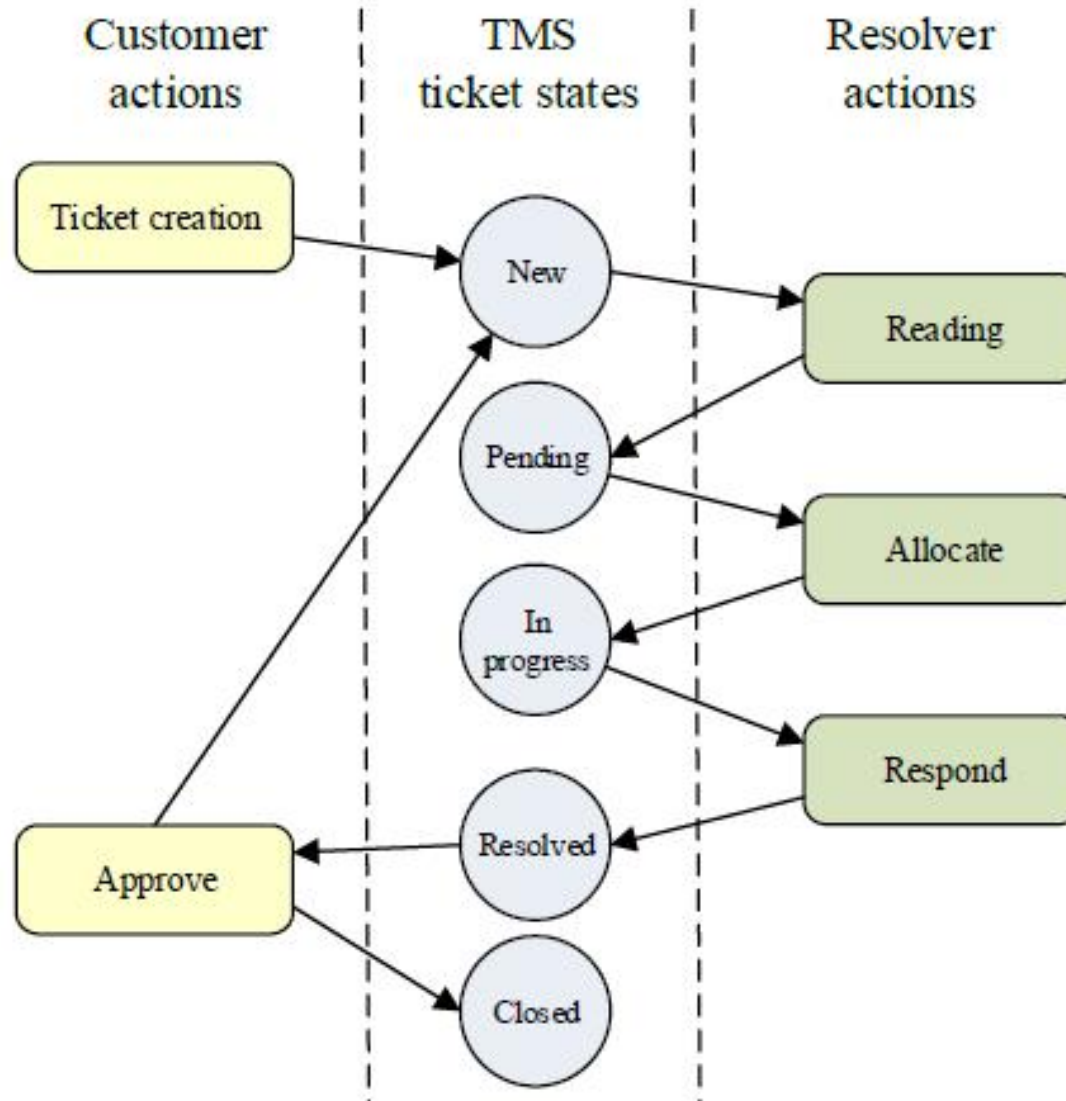
- **Approving** - CU approves or rejects the response. TMS receives those details and updates the database.
- **Creating test cases** - CU creates test cases for review of functionalities. TMS stores those details and available to all interested parties.
- **Summary reports** - A user sends a request to TMS and receives a specific type of report with requested information.

# Functional description (2)

- **Detailed reports** - A user sends a request to TMS with a specification of details and receives a detailed report.
- **Site analytics** - A user sends a request to TMS for site analytics and receives appropriate content as response.
- **Email notification** - A user sends a request to TMS for a change in a ticket state as an e-mail identification.



# Activity description and process roles





# SaaS solution - Basic architecture modules

- System Management Module (SMM)
- Company Management Module (CMM)
- Core functions module (CFM)
- Additional Functional Module (AFM)

# SaaS solution - Basic architecture modules

- **System Management Module (SMM)**
  - general for all companies
  - manages all resource provisioning in the cloud
  - provide user management, company subscription management, authentication, authorization etc.
- **Company Management Module (CMM)**
  - Company specific module
  - responsible for all general configurations within a company
  - Defining relationships with the customers
  - providing user management, customer management, project and module management, etc.

# SaaS solution - Basic architecture modules

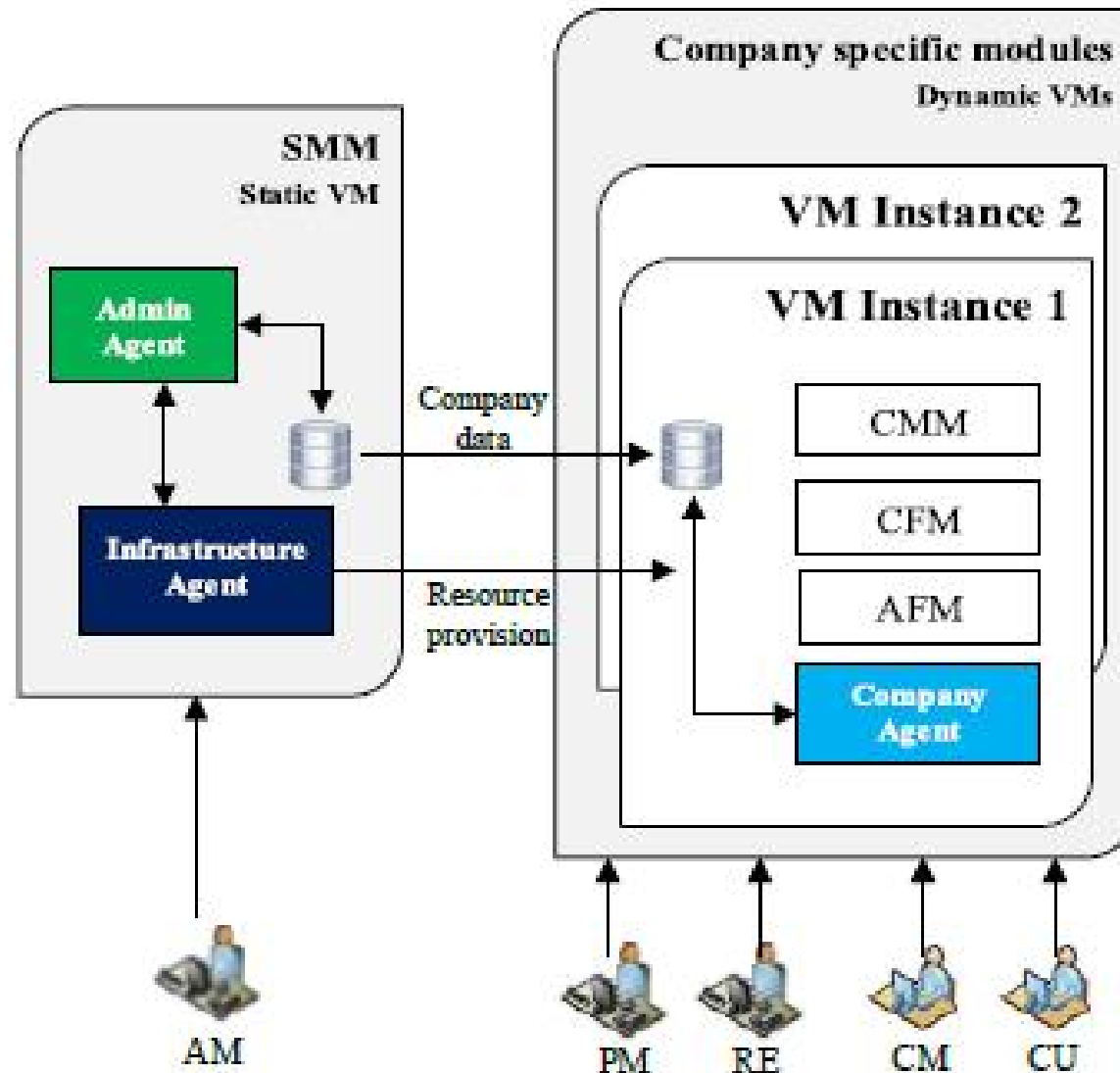
- **Core functions module (CFM)**
  - Company specific module
  - responsible for all essential process to realize ticket management (bug reporting) within a company
  - Enables a connection between companies (providers) and their customers
  - provides functionalities such as: ticket creation, responding by a resolver, customer approving of the response, etc.
- **Additional Functional Module (AFM)**
  - dedicated for add-on functions, such as enabling a possibility to create and execute test cases, especially useful for bug reporting systems.

# Classification and clustering into modules

Functionality	Static / Dynamic	Module
registration	Static	SMM
customer admin	Static	CMM
resolver admin	Static	CMM
ticket issuing	Dynamic	CFM
ticket reading	Dynamic	CFM
allocating resolvers	Dynamic	CFM
resolving	Dynamic	CFM
approving	Dynamic	CFM
test-case creation	Static	AFM
test-case executing	Dynamic	AFM
reporting	Static	CMM
notification	Dynamic	CFM



# SaaS solution - Optimized cloud organization



# Conclusion

- We have developed a new cloud based system that works in optimized mode
- Our design uses a static VM hosting the SMM with Admin and Infrastructure Agents.
- The dynamic VMs instances are invoked whenever the admin module triggers a need. Each dynamic VM consists of three modules CMM, CFM and AFM and the Company agent.

# Future work

- The purpose of developing this type of software or introducing application is to realize a web service for ticket management (bug reporting).
- In the future we plan to develop a SaaS solution that deals with these issues and open questions, and realize series of experiments to analyze the performance and the cost of the new cloud based TMS concept.