

# Process Remixes - Mixing Legacy with Process Orchestration



**Sukriti Goel**, Jyoti M. Bhat  
BPM Research Group  
Software Engineering and  
Technology Labs  
Infosys Technologies Limited

**Infosys**<sup>®</sup>  
POWERED BY INTELLECT  
DRIVEN BY VALUES

BPM '10 Conference, Hoboken, NJ, USA, Sept 2010

# Agenda

- Problem Statement
- Introduction of Process Remix
- Case Study 1
  - Order Provisioning System of a Telecom Company
- Case Study 2
  - Large Global IT Services Organization
- Related Technologies

# Problem Statement

Need for quick turn-around time constantly changing processes to maintain competitive edge

- To capture more information
- To comply with ever changing regulations
- For improving the process efficiency

Need to monitor end to end process

Organizations have already invested in creating and maintaining systems over years, it is expensive or complex to update or replace those systems

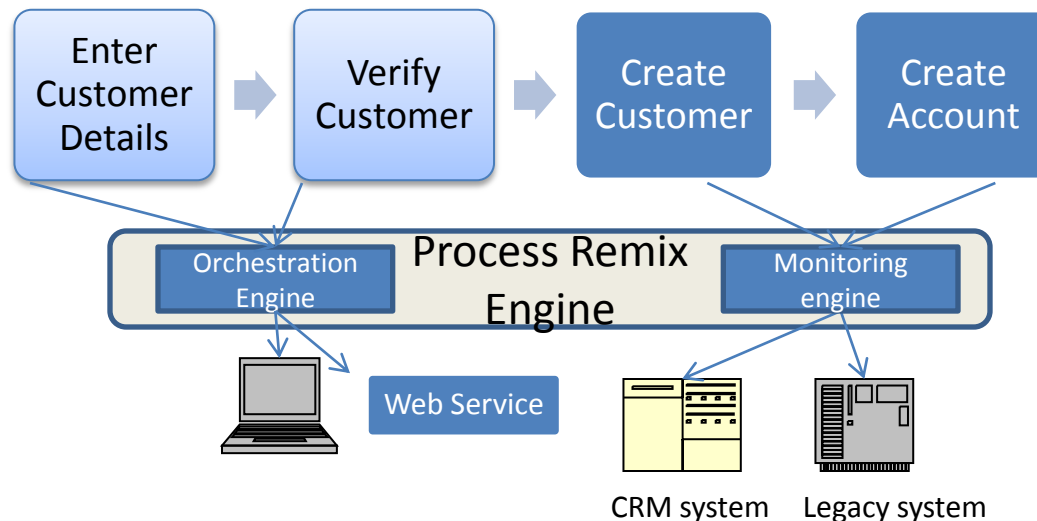
# What is Process Remix

Process remixes are achieved by using **process orchestration** capabilities with **non-intrusive monitoring** of the existing systems

The process model in process remixes consists of two types of activities.

- Activities that are to be added to the current process related to information capture, approvals, etc, which are orchestrated by the Process Remix engine.
- Activities of the type '**monitored activity**' which are functionalities that are executed in various other applications and monitored by the monitoring engine and tracked by the Process Remix engine.

Alternate version of the process is created that implements the requirements of the business users related to process visibility and business user productivity.



# Why Process Remix

Can monitor same process from perspectives of various stakeholders by changing the configuration.

- Provides higher abstraction of process information to top management
- Provides day to day operation dashboard to Operation Manager
- Provides data for process analytics for process improvement to process owner

Can extend/modify the process in the current systems with minimal IT effort

# Process Remix

## How

- The status of activities executing in existing systems is tracked while the new activities are orchestrated by the Process Remix engine
- Any monitoring mechanism (intrusive/non-intrusive) can be used
- Both type of activities can be allocated to users using the task allocation mechanism (similar to BPMS capabilities)

## Pros

- Process variations can be created with almost no IT effort
- Provides end-to-end view of the process instances
- Can help organizations modify and monitor their process without changing code in existing applications.

## Cons

- Process Remix may provide near real time status as compared to real time status
- Process engine is dependent on a monitoring mechanism to track the activity status

# How Process Remix Works

## Model

- Model the activities to be **orchestrated** as regular activities
- Model the activities executing in existing systems as **monitored** activities
- Allocate all manual activities including monitored activities to user using **task allocation rules**

## Deploy and Configure

- **Deploy the process** on Process Remix engine
- Configure the monitored engine
- Set performance indicators for activities

## Execute

- Process Remix engine **orchestrates** the regular activities and **maintains a waiting list** of monitored activities
- **User Inbox** is populated with the tasks waiting for his/her actions
- Monitoring engine **tracks the status** of monitored activities
- Process Remix engine **correlates the monitored activities** to the process instances

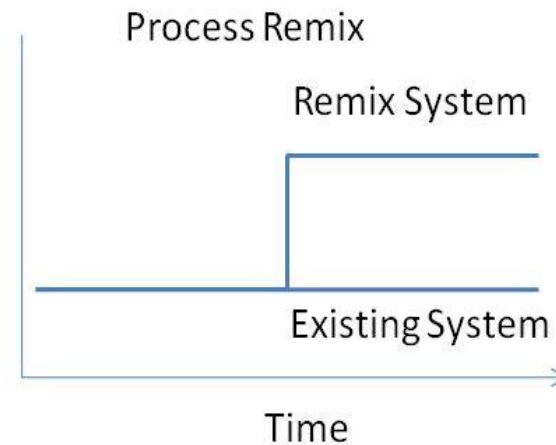
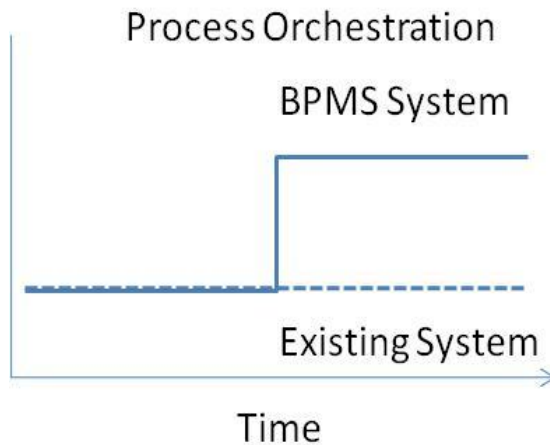
## Monitor

- Process remix engine maintains the **process execution details** for both orchestrated and monitored activities
- **Alerts** are sent to users as configured
- Process managers are provided with **dashboard** to oversee the process progress

# BPMS Vs. Process Remix

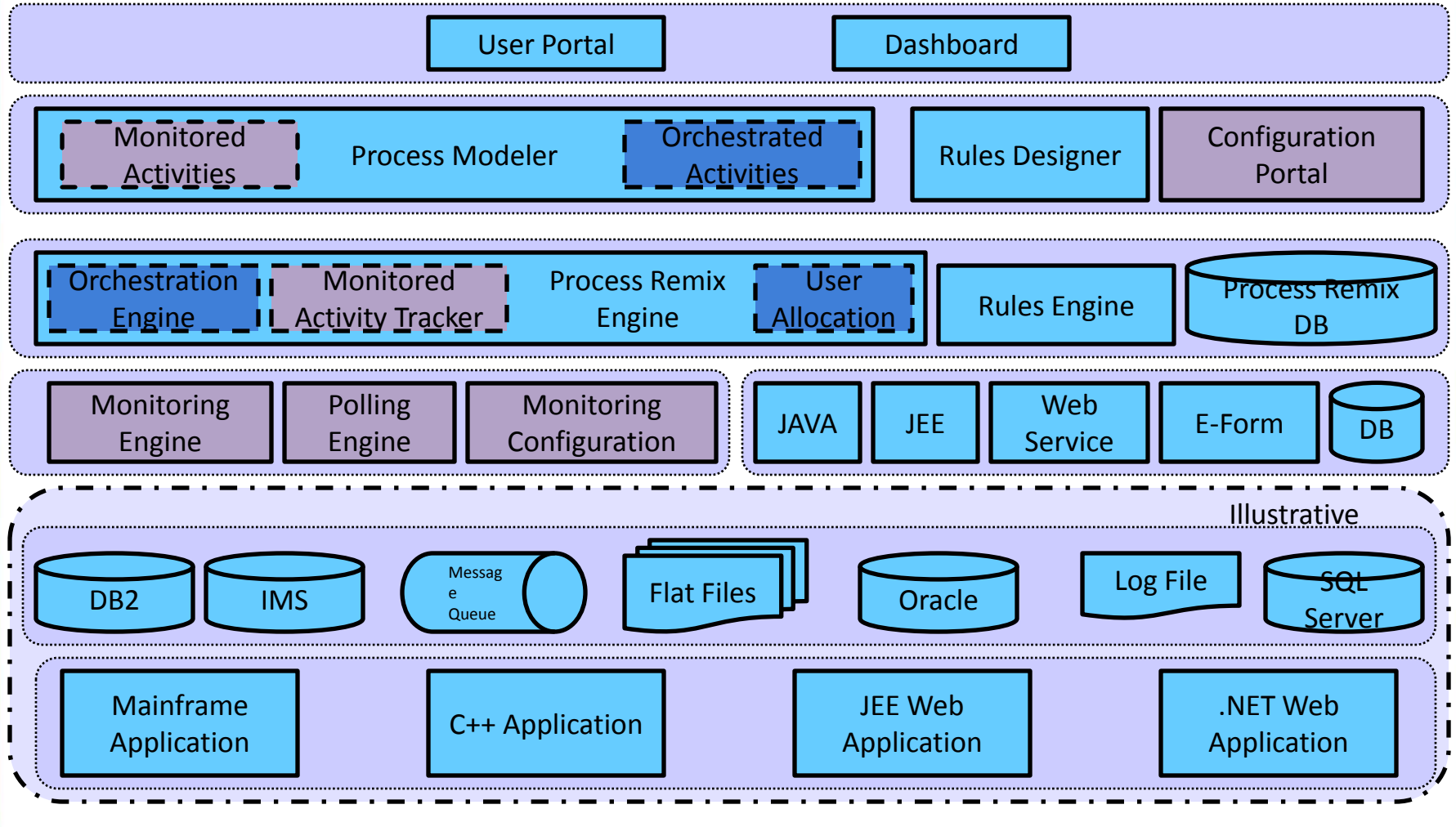
BPMS uses adaptors and availability of services to orchestrate process executing in existing heterogeneous systems while Process Remix uses process monitoring tools

Existing systems and processes cease to exist once BPMS is implement while Process Remix can co-exist with existing processes and systems





# Process Remix Building Blocks



# Case Study 1 : Order Provisioning System of a Telecom Company

# Case Study

## Organization

- Large Telecommunication Organization

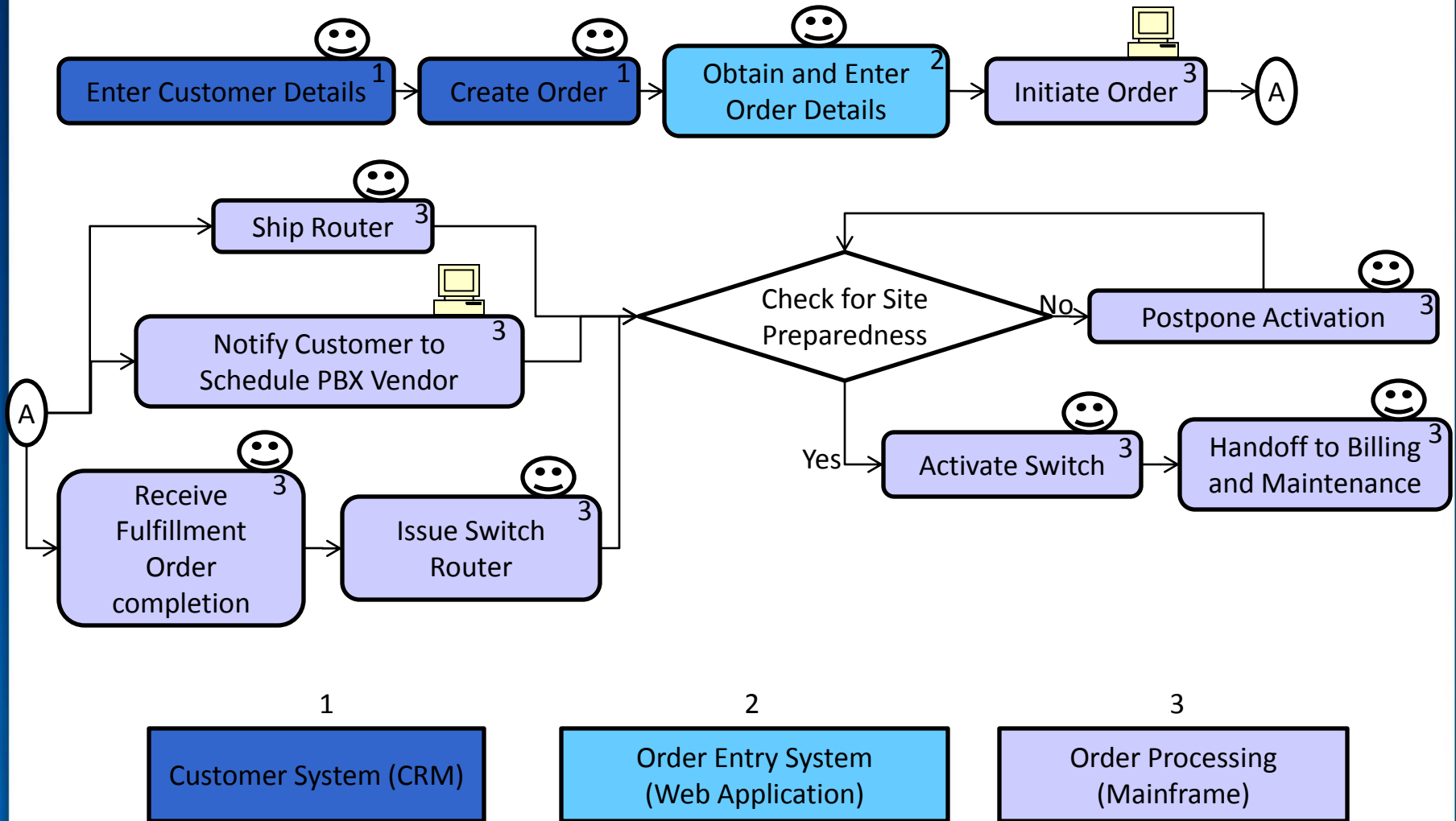
## Process

- Order Provisioning System for Voice over IP for businesses

## Current Technology

- Multiple types of systems - web enabled data entry systems, legacy order processing systems, etc.
- The integration across systems is manual.

# Current Process Details



# Process Requirements

Traceability of order

Eliminate need to login in to multiple systems

Consolidated Inbox where all the activities to be performed by users are maintained

Notification in case of delay in any activity

Automated allocation of an order to an order manager who is responsible for order completion

Reassign of an order in case an order manager is not available

Business users want to capture additional details during the process execution.

- In case there is a delay in activities 'Receive Fulfillment Order completion', 'Ship Router' and or in site readiness they wanted to capture the reason of delay.

Stakeholders did not want to make changes in multiple systems

# Available Options

## BPMS implementation for the entire process

- All the participating system would need to change to implement BPMS

## Partial BPMS implementation where process continues to execute in heterogeneous systems, BPMS is used to allocate tasks and maintains process logs

- Once the task is completed by the user, user comes back and update the BPMS about the completion of task
- New activities can be added to the process with ease, e.g., activity to capture reason for delay can be added easily
- No change needed in existing systems
- The process execution data would corrupt in case user does not update the BPMS immediately after the completion

## How We Implemented Process Remix

# Process Remix Solution

Process model was created using Process Remix modeler, the model consists of

- Orchestrated activities such as 'Enter reason for Delay'
- Monitored activities such as 'Receive Fulfillment Order Completion'

Task allocation rules for all manual activities such as 'Ship Router' assigned to 'Order Manager'

New code was created for

- UI screens of new orchestrated activities such as 'Enter Reason for Delay'
- New automated services such as 'Notify Customer to Schedule PBX Vendor'

For each monitored activity, the data event was identified and configured – 'Enter Order' mapped to 'OrderDetails' table in CRM system

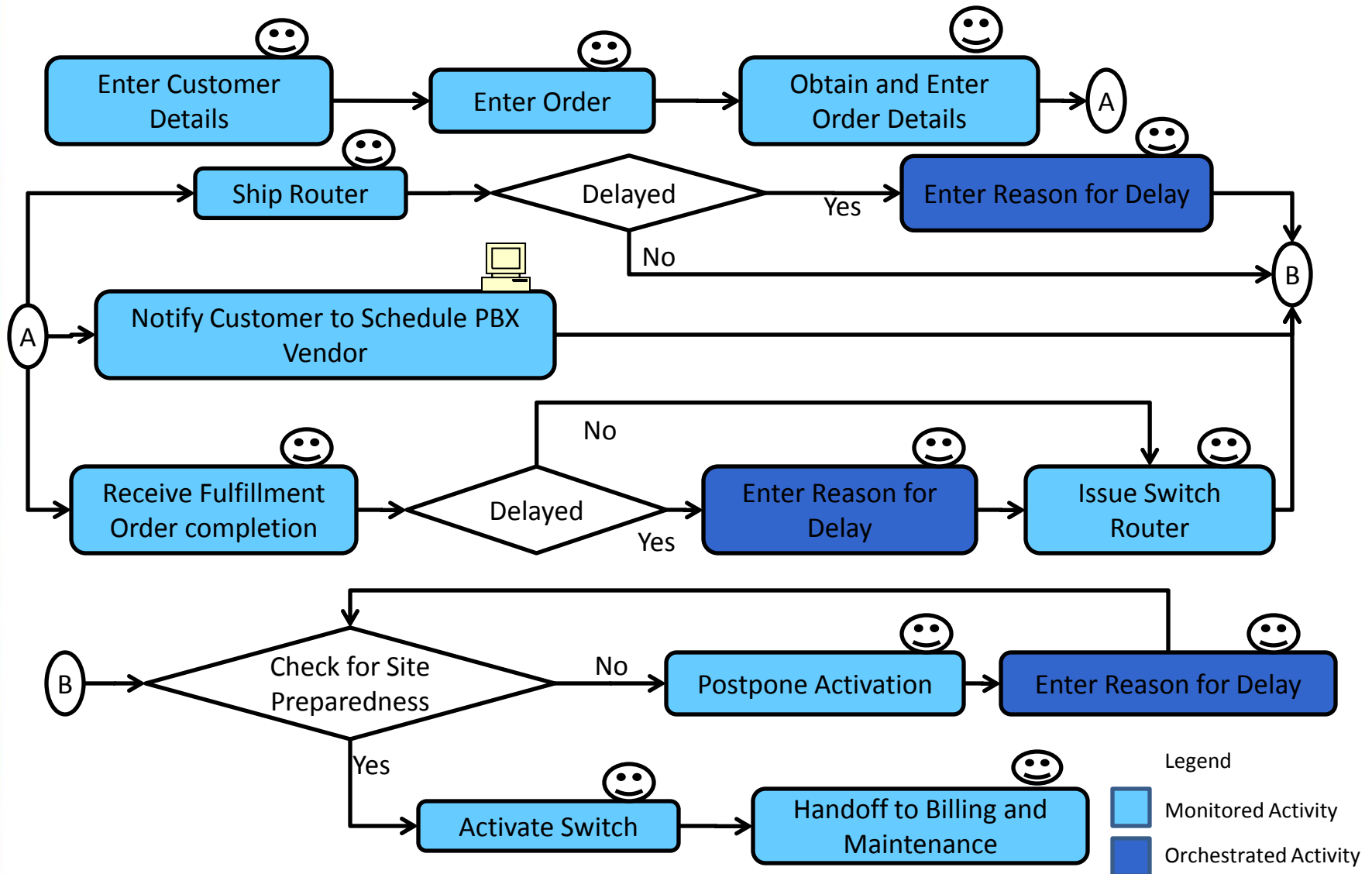
SLA and alert configuration was done – 'Activate Switch'

## Advantages

- New activities e.g. activity to capture the reason for delay was added to the process with ease
- IT team did not have to change the existing systems
- The order status is visible to all stakeholders
- Process monitoring is available from different perspectives i.e. Order Manager, Senior Management and Process Owner



# To-Be Process Model



## Case Study 2: Employee On-boarding of a Large Global Organization

# Case Study 2: IT Services Organization

## Organization

- Large Global IT services organization with 100,000+ employees

## Objective

- Streamline the employee on-boarding process

## Challenges

- Employee has to complete lots of activities in systems of various functions and departments after joining, there is no defined process
- Some of these activities are dependent on actions by someone else
- HR needs traceability of what all activities completed by employee

## Case Study 2: Process Remix Solution

Modeled the employee on-boarding process

- Added the order to the activities wherever needed
- Activities executed in various systems were modeled as monitored activities
- Activities were allocated using task allocation rules

Once deployed the process was configured to read the status of activities from various data sources

SLA configuration was done for process and individual activities

## Case Study 2: Execution

Process was started on the process remix engine for monitoring

Employees are provided with task list of all the tasks to be completed in various system. Each task is provided with a link to go the specific system to complete the task

As employees complete the task, the status is updated in process remix engine with the task completion timestamp

Alerts are sent to user in case SLA is breached

HR manager can see the status of on-boarding process for each new employee

## Other Related Technologies

# Related Technologies

Business Process Management System (BPMS)

Service Oriented Architecture (SOA)

Process Monitoring Solutions

Business Process Mashup

Using BPEL constructs (Receive/Pick)

# Q n A!



Sukriti Goel

Product Architect

[Sukriti\\_Goel@infosys.com](mailto:Sukriti_Goel@infosys.com)

Jyoti M. Bhat

Principal Research Scientist

[jyotimb@infosys.com](mailto:jyotimb@infosys.com)

**Infosys**<sup>®</sup>  
POWERED BY INTELLECT  
DRIVEN BY VALUES



# Business Process Management Systems (BPMS)

## What

- Process and Rules is outside the applications
- What you model is what you execute

## Why

- Need to change processes quickly
- Need to monitor processes as they execute
- Provide capability of changing business processes to process owners

## How

- Use technologies like Workflow systems, EAI, SOA

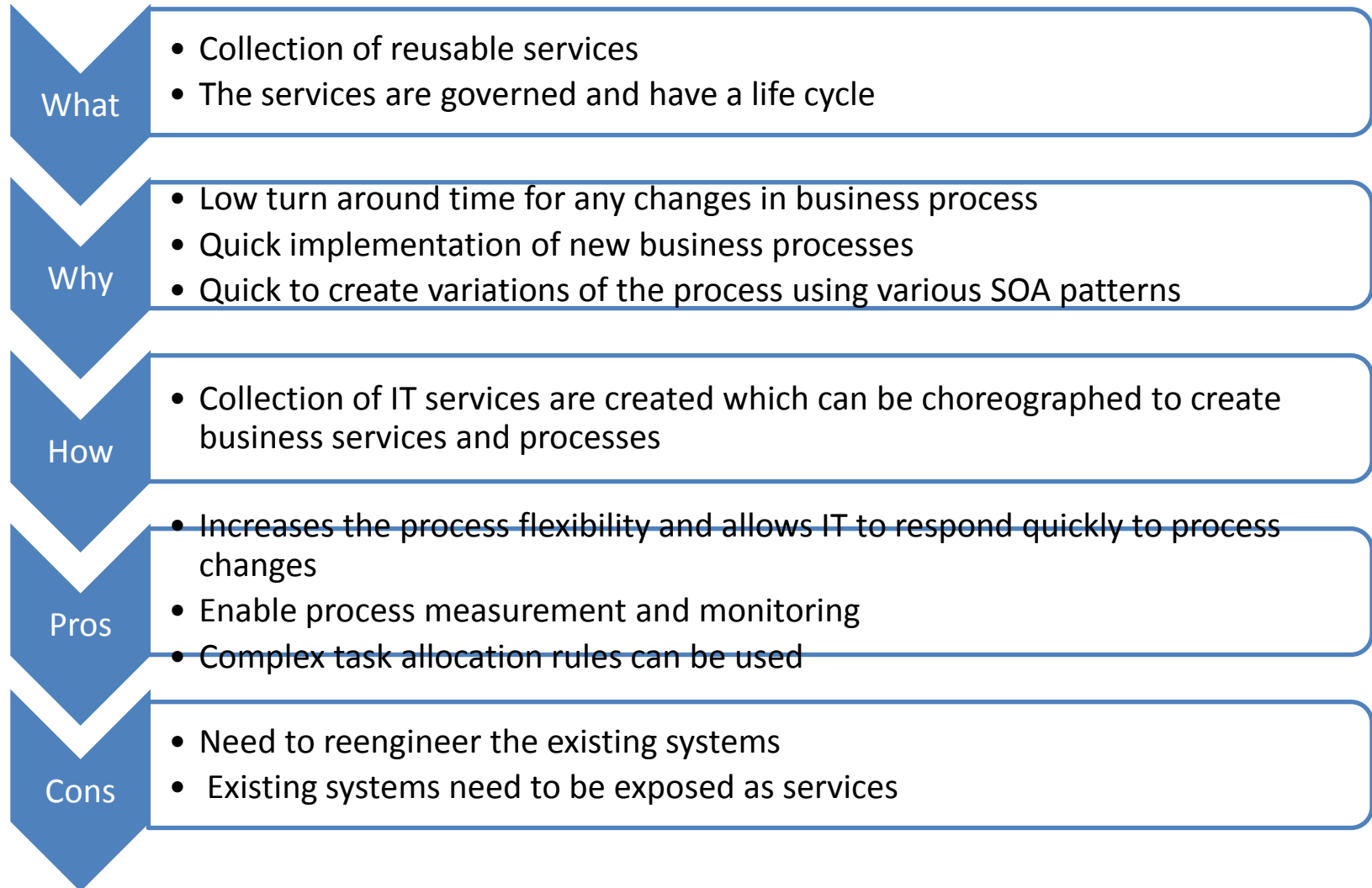
## Pros

- Increases the process flexibility and allows IT to respond quickly to process changes
- Enable process measurement and monitoring
- Complex task allocation rules can be used

## Cons

- Need to reengineer the existing systems
- Existing systems need to be exposed as services

# Service Oriented Architecture (SOA)



# Process Monitoring

## What

- Enables process monitoring as process executes across systems
- Provides alerts and dashboard capability

## Why

- Business users can improve the process based on the process reports
- Operation Manager can handle operations better with alert and dashboard capability

## How

- Uses persisted data to find the process progress
- Uses probes in the middleware, network layer, etc to find process progress
- Provide data in format provided by tool vendor to generate reports

## Pros

- Less effort and minimal code to use this method

## Cons

- Does not provide process flexibility

# Business Process Mashup

## What

- Enables process monitoring as process executes across systems
- Provides alerts and dashboard capability

## Why

- Business users can improve the process based on the process reports
- Operation Manager can handle operations better with alert and dashboard capability

## How

- Create: Business analysts compose business processes that combines existing content, applications, and services.
- Use: These business processes to Mashup server that executes the process by routing information between different people and systems

## Pros

- Very easy to implement in case functionality is available as services
- Most tools provide UIs enriched with data from multiple sources, thus making decision making easy

## Cons

- Extensively depends on availability of functionality as services to deliver its promise
- If services are not available, the effort is much higher

# Using BPEL constructs (Receive/Pick)

## What

- Use of BPEL functionality

## Why

- Business user can view the process status as executed in BPMS and in external systems
- Business user can be alerted if there is a delay in execution of task in external systems

## How

- Activities to be monitored can be modeled as Receive/Pick construct of BPEL
- Applications have to be modified to notify the process engine

## Pros

- Provides end-to-end view of the process instances
- Process modeling is easy

## Cons

- External systems need to be changed to send notification to the process engine
- The task modeled as BPEL constructs cannot be allocated to a users