

RED HAT  
**SUMMIT**

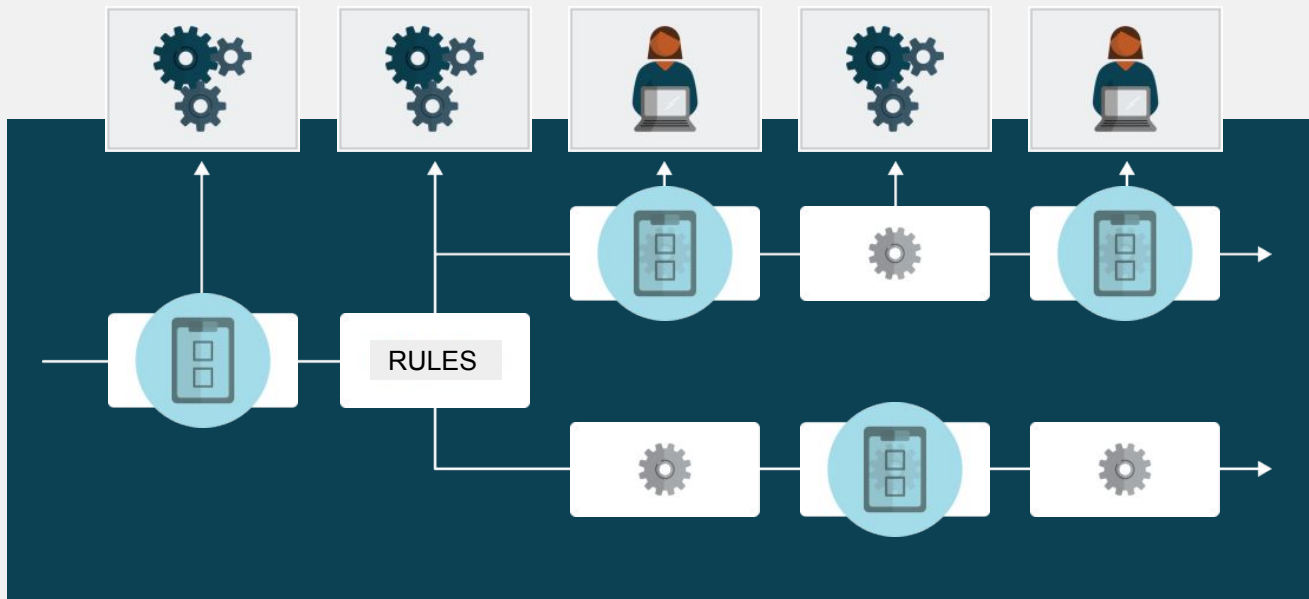
# TRANSFORMING A PROCESS-DRIVEN ENTERPRISE TO A DIGITALLY ENGAGED ENTERPRISE

Prakash Aradhya, Director of Product Management, Red Hat  
Phil Simpson, Product Marketing Manager, Red Hat  
Jeanine Bradley, Sr. Manager, Railinc

May 2, 2017

# PROCESS-DRIVEN ENTERPRISE

BPM technology has enabled automated back-office operations...



- ✓ Visibility
- ✓ Agility
- ✓ Consistency

# BUT THE WORLD IS CHANGING...

“...now we have to compete against new disruptive startups that are hiring kids fresh out of college drinking loads of Mountain Dew!”

(CIO @ Global Insurer)

# CASE IN POINT:



# THEN THIS HAPPENS...



Dollar Shave Club was founded 110  
years after Gillette...

Dollar Shave Club was founded 110  
years after Gillette...

Five years later they sold to Unilever for  
a reported \$1Bn in cash...

Dollar Shave Club was founded 110  
years after Gillette...

Five years later they sold to Unilever for  
a reported \$1Bn in cash...

Gillette responded by digitally  
transforming, starting an online business...



Dollar Shave Club was founded 110  
years after Gillette...

Five years later they sold to Unilever for  
a reported \$1Bn in cash...

Gillette responded by digitally  
transforming, starting an online business...

Today Gillette claims 21% of the  
online shaving market

# DIGITAL TRANSFORMATION

“While its definition is fluid, ‘digital transformation’ is commonly understood as the means to exploit emerging technology to improve the customer experience and enable more adaptive business operations.”

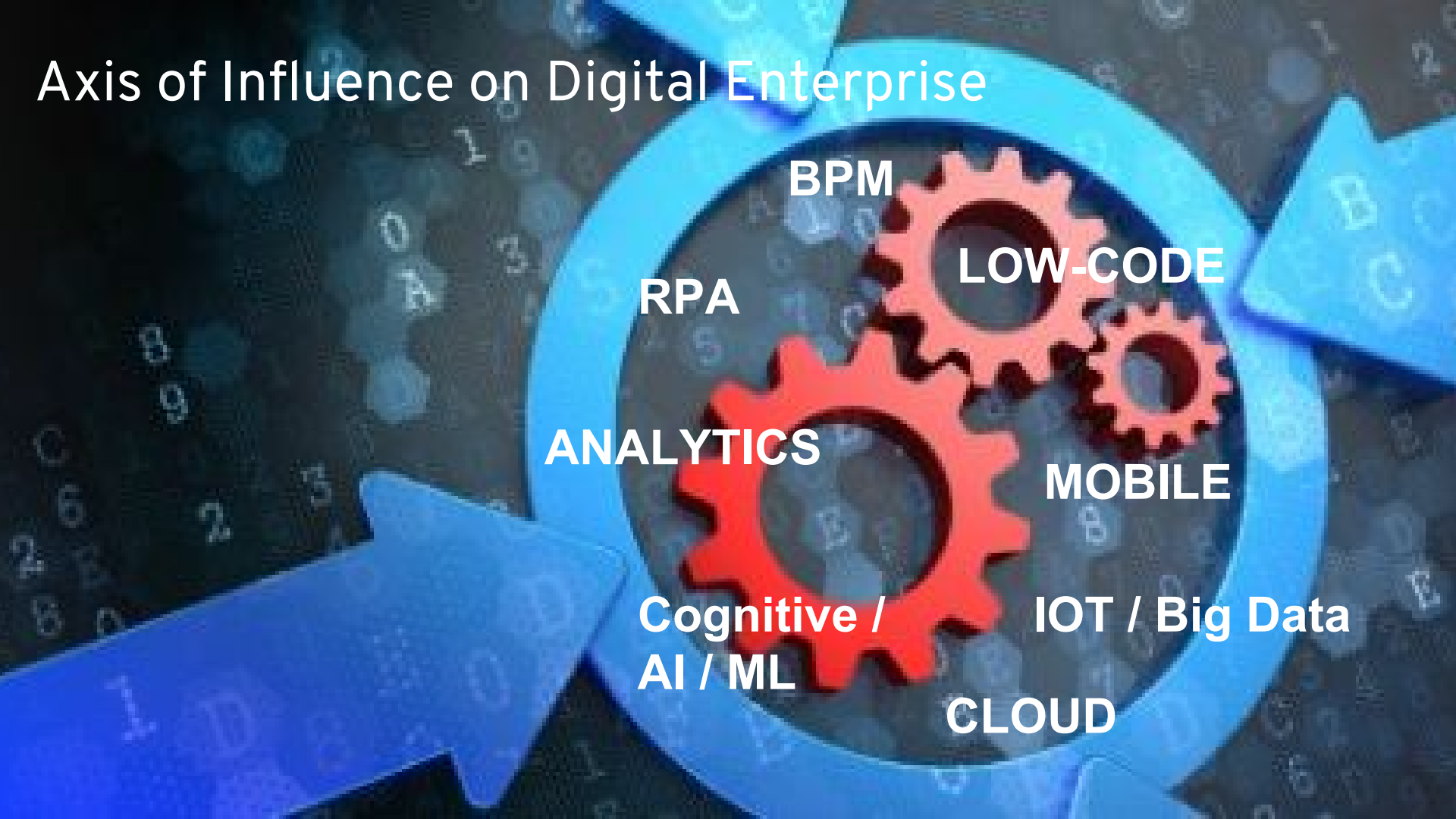
- Carl Lehmann, Principal Analyst, The 451 Group

# DIGITAL TRANSFORMATION

“But digital transformation is also something more. Changing, improving and adapting how customers engage and interact with an enterprise often impacts how **business processes** are designed, how the **workforce collaborates** in response, and how business partners in kind must react to new opportunity and **ways of doing business.**”

- Carl Lehmann, Principal Analyst, The 451 Group

# Axis of Influence on Digital Enterprise



RPA

**Robotic Process  
Automation**

# Will Robots Replace Human Jobs ?



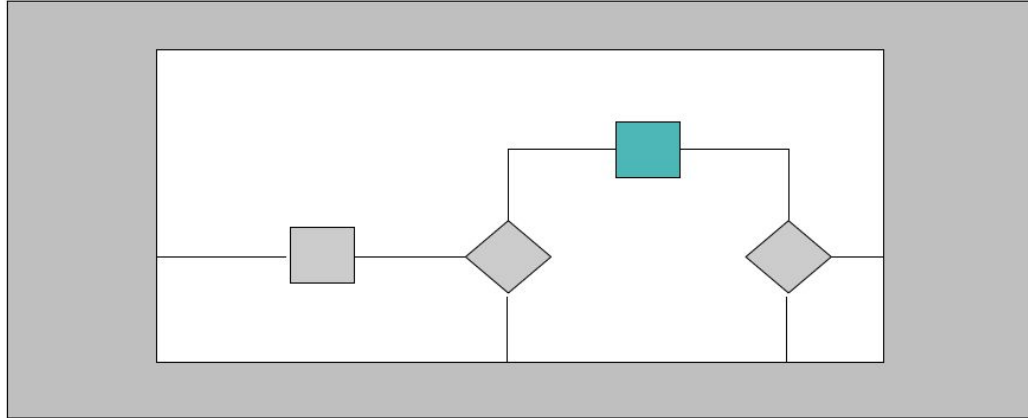
Drivers, Doctors,  
Pharmacists, Teachers  
....And even Soldiers

Needed anymore ?

‘Robots will take over the World in 100 years’ - Stephen Hawking

‘Universal Basic Income will be a real thing ‘ - Elon Musk

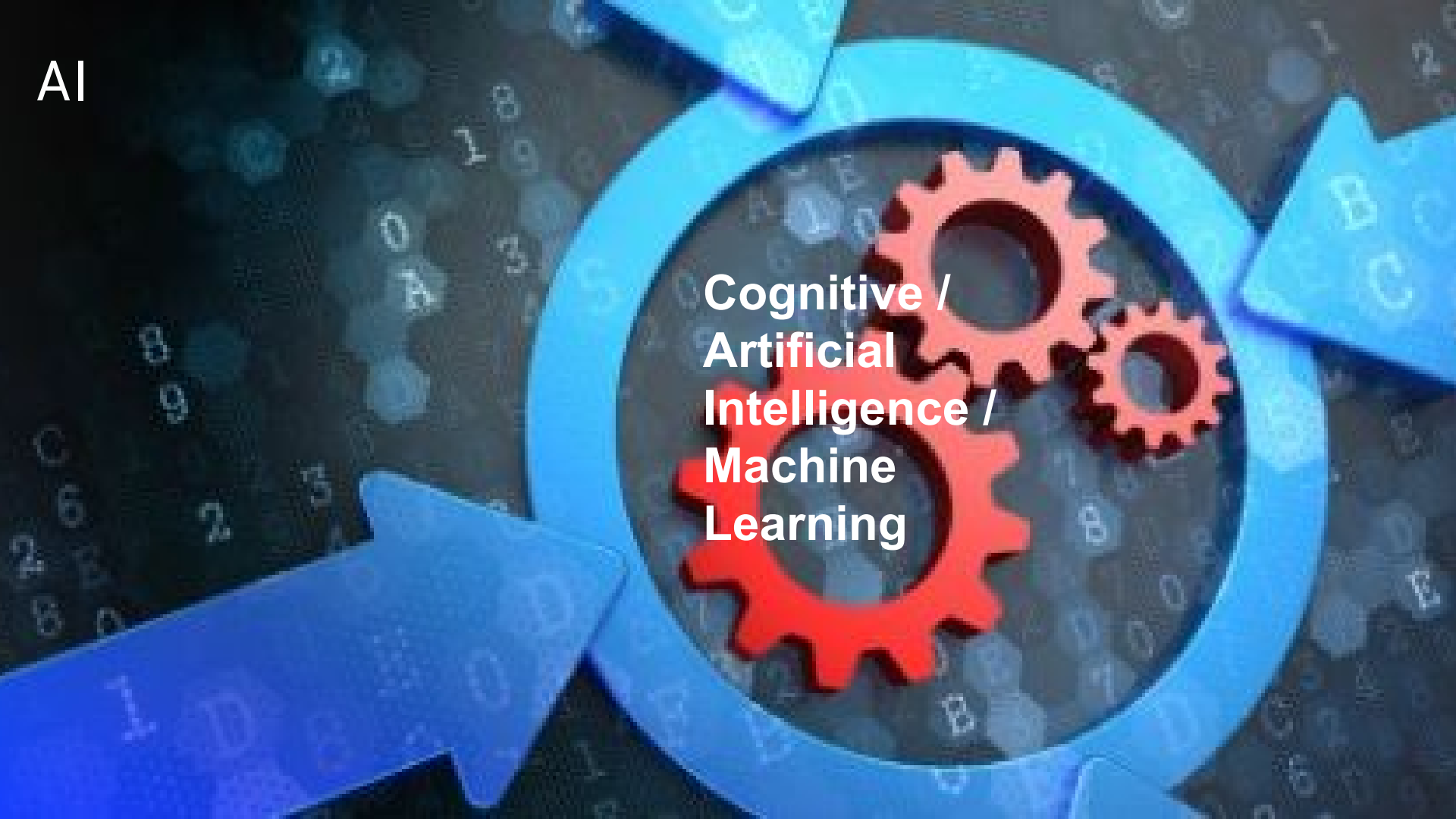
# Work item Automation



Automate clearly defined, repetitive and rule based human work items

Examples: Amazon order fulfilment, Statement reconciliation, Claim processing, Compliance reporting, profile updates

AI



**Cognitive /  
Artificial  
Intelligence /  
Machine  
Learning**



Lowcode



Agile  
App Dev for  
Citizen  
Developers

IOT / Bigdata / Analytics



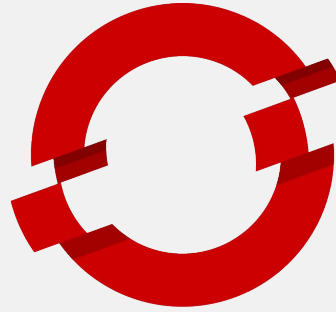
**Intelligent  
Decision  
Making  
And Reporting**

Cloud

Hosted  
Platform

The image features a central blue circle containing three interlocking gears: one large red gear and two smaller orange gears. A large blue arrow points from the bottom left towards this circle. The background is dark with faint, scattered binary code (0s and 1s) and letters (A, B, C, D, E). The text 'Cloud' is in the top left, and 'Hosted Platform' is centered over the gears.

# KEY TO DIGITAL PROCESS AUTOMATION



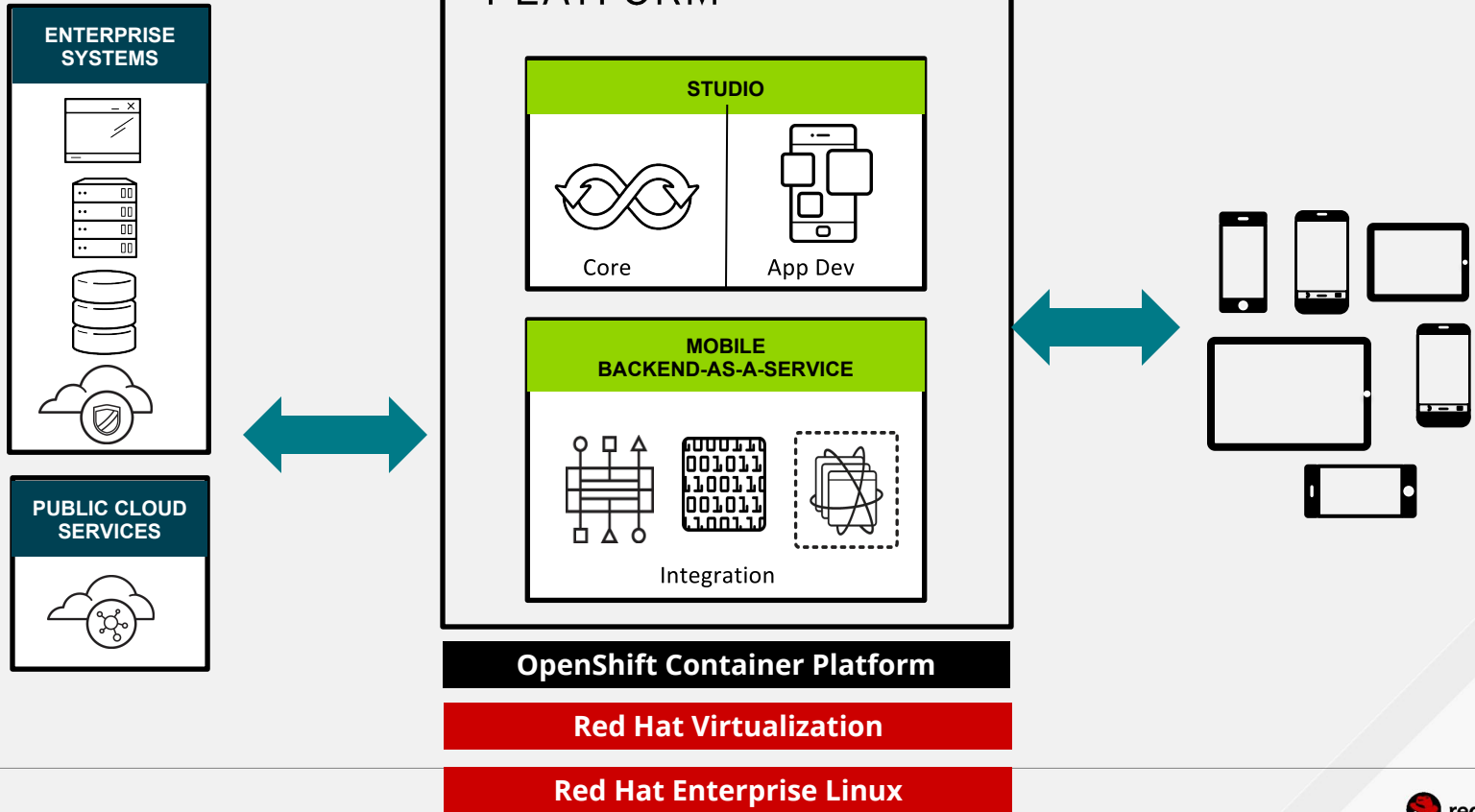
**RED HAT**<sup>®</sup>  
OPENSIFT

Mobile

Multi Platform  
Applications

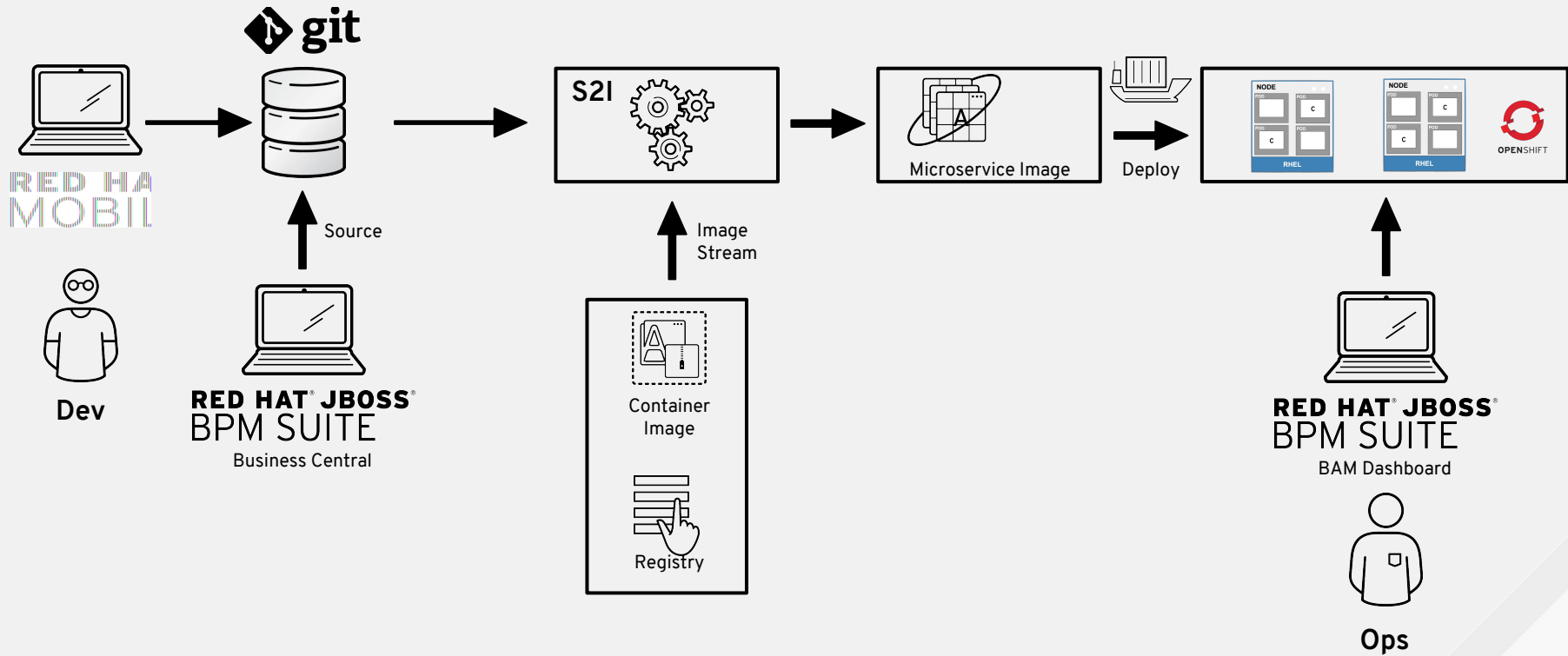
The image features a central graphic consisting of a large blue ring with four blue arrows pointing inward from the top, bottom, left, and right. Inside this ring are three interlocking gears: one large red gear at the bottom and two smaller red gears at the top. The text 'Multi Platform Applications' is centered over the gears. The background is dark with faint, scattered binary code (0s and 1s) and letters (A, B, C, D, E).

# RED HAT MOBILE



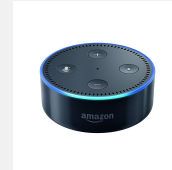
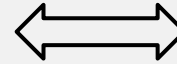
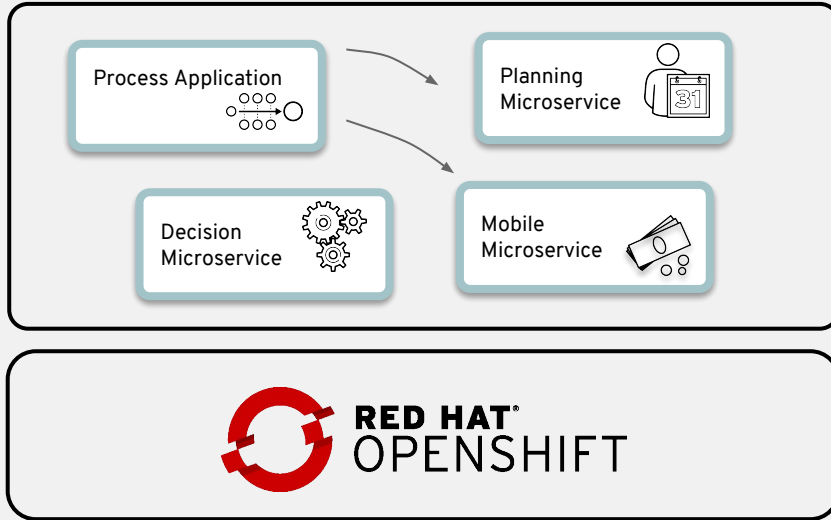
# BUILDING DPA APPS

With JBoss BPM Suite, Red Hat Mobile and OpenShift



# RUNNING DPA APPS

Modern  
microservices  
architecture



Next-gen user  
experience





# EXAMPLE USE CASES

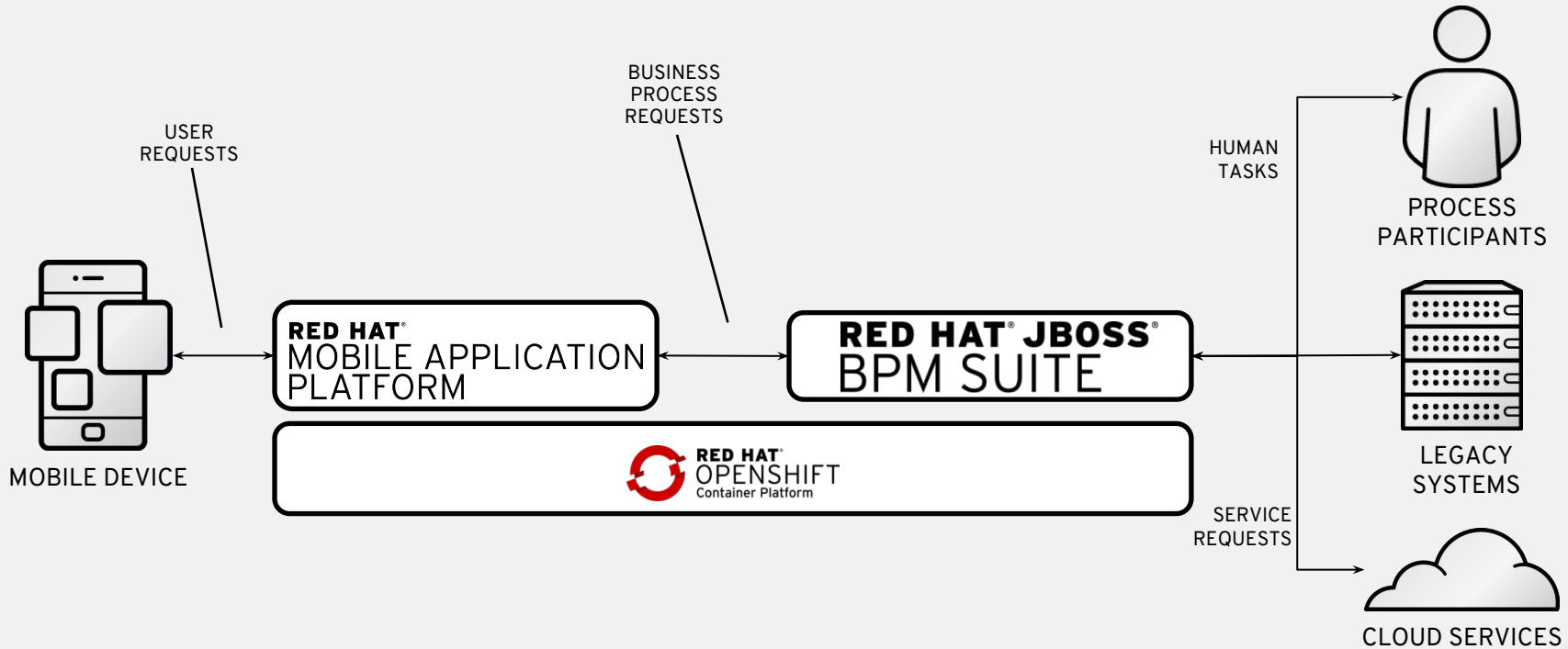
# SELF-SERVICE APPLICATIONS



## INSURANCE CLAIMS PROCESSING

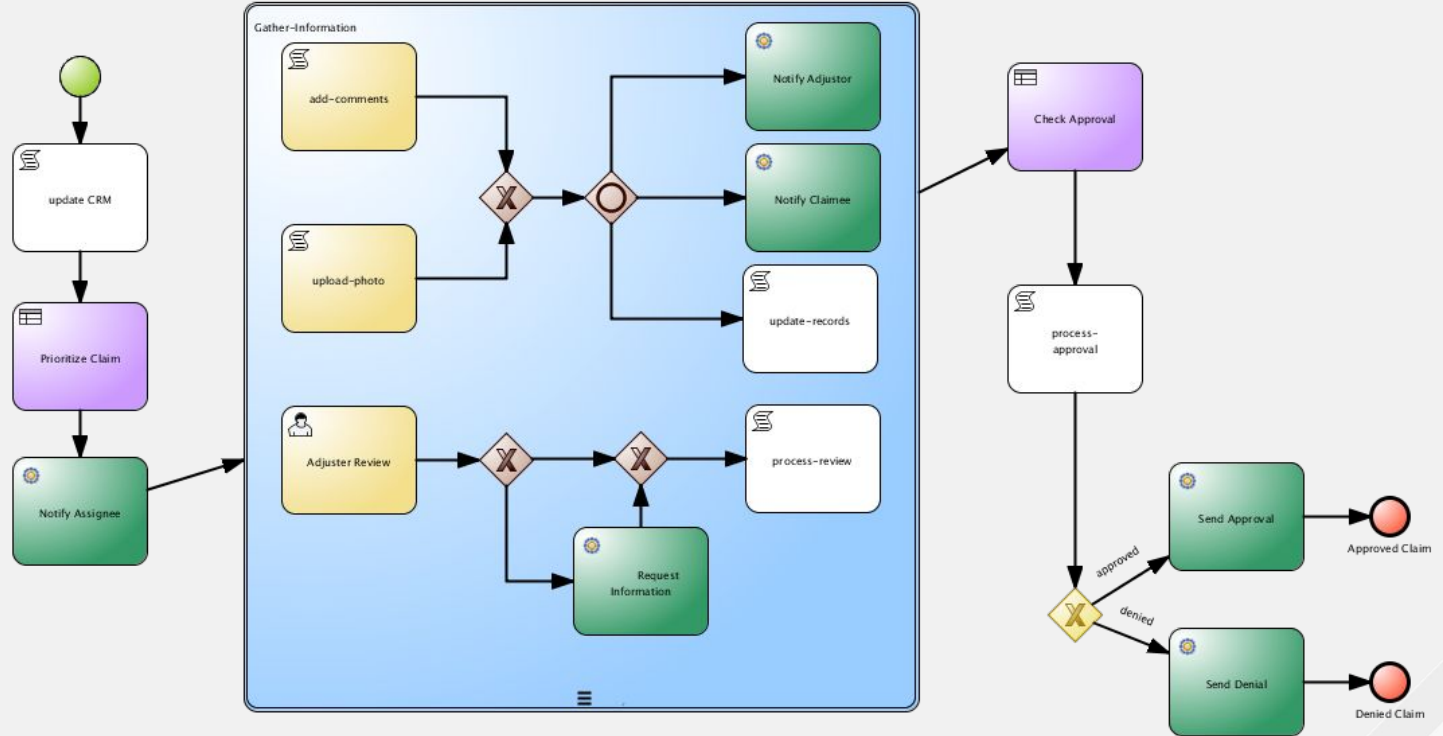
- **Reduce** claims processing cycle and costs
- **Improve** data capture
- **Eliminate** manual or paper-based processes
- **Streamline** workflow

# SYSTEM ARCHITECTURE



# INSURANCE CLAIM PROCESS

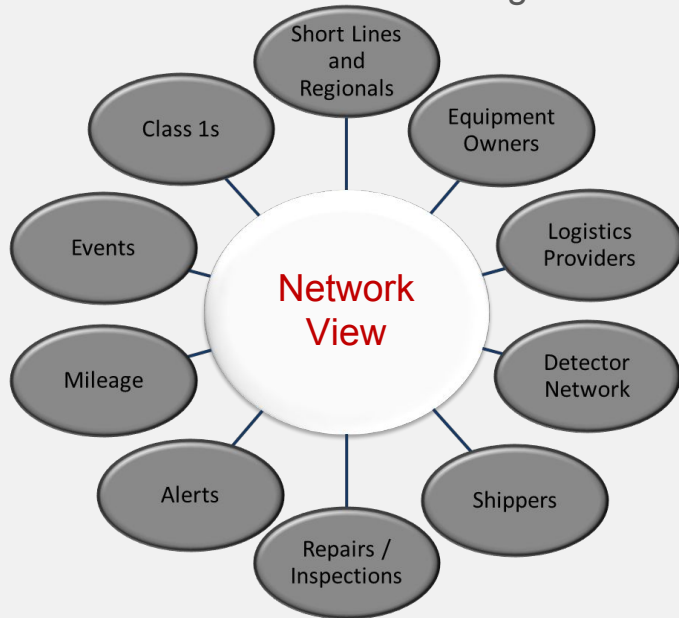
mobile-claim-process v.1.0 (mobile-claims-bpm.mobile-claim-process)



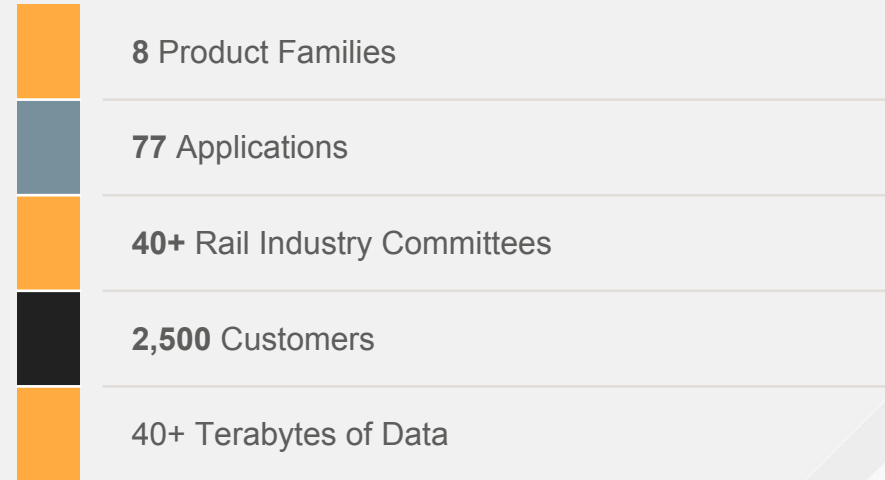
# RAILINC

Software as a Service / Data as a Service / Value Added Network

Single Industry Source for Network  
Freight Rail Information



## Railinc by the Numbers



# Rules and Process Management Initiative

Implementation of a formal approach that guides the identification, definition, implementation and management of business rules and processes.

Initiative Benefits;

Increase Railinc **Agility**

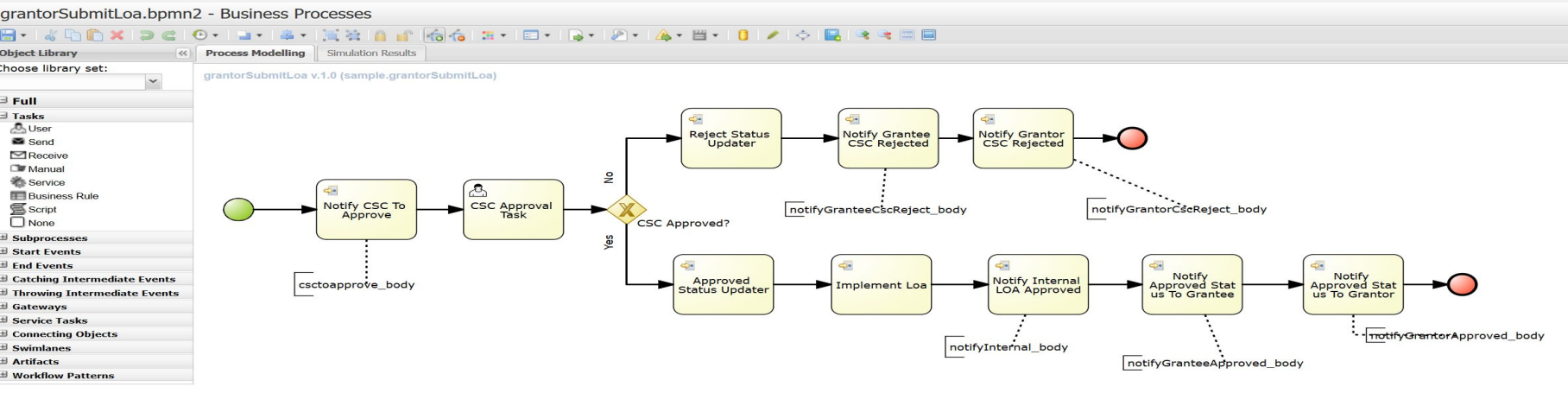
Expand Railinc's **Industry Expertise**

Increase **Product Quality**

Reduce **Total Cost of Ownership**

# DAMAGED & DEFECTIVE CAR TRACKING APPLICATION

## Example Business Process (BPMN)

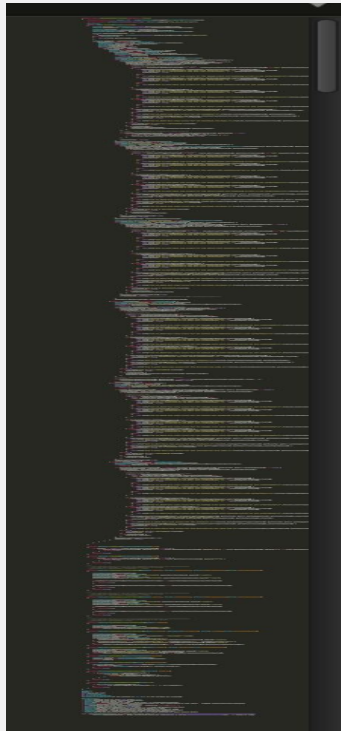


### Benefits

- Responsiveness/Agility; added 2 additional workflows without increased budget or delayed timeline
- Knowledge; IT PM, Developers and Business Analysts all changed over twice without impact
- Continuous Improvement; business monitoring capability facilitates discussions with committee on improvements

# RAILINC APPLICATION (RED HAT JBOSS BPM/BRMS)

## Code Quality/Simplicity



Note the scroll bar

```
// If incident is a damage
if (isDamage(railEvent.getNetworkEvent())) {
    if (hasViewAccessForDamage(subject, carMarkOwner, handlingCompanyId,selectedCompanyId)) {
        LOG.info("Uid {},<< Permitted to see DDCT with handling carrier >>" + handlingCompanyId,transId);
        LOG.info("Uid {},<< Permitted to see DDCT with carMarkOwner >>" + carMarkOwner,transId);
        isPermitted = true;
    }else
    {
        LOG.info("Uid {},Dont have view access for the damage event with handling carrier {} and carMarkOwner {}",transId,handlingCompanyId,carMarkOwner);
    }
} else if (isDefect(railEvent.getNetworkEvent())) {
    if (hasViewAccessForDefect(subject, carMarkOwner, handlingCompanyId,selectedCompanyId)) {
        LOG.info("Uid {},<< Permitted to see DDCT with handling carrier >>" + handlingCompanyId,transId);
        LOG.info("Uid {},<< Permitted to see DDCT with carMarkOwner >>" + carMarkOwner,transId);
        isPermitted = true;
    }else
    {
        LOG.info("Uid {},Dont have view access for the defect with handling carrier {} and carMarkOwner {}",transId,handlingCompanyId,carMarkOwner);
    }
}
}
```

Imperative; have to determine what each item is;  
looping; bunches of 'if' statements

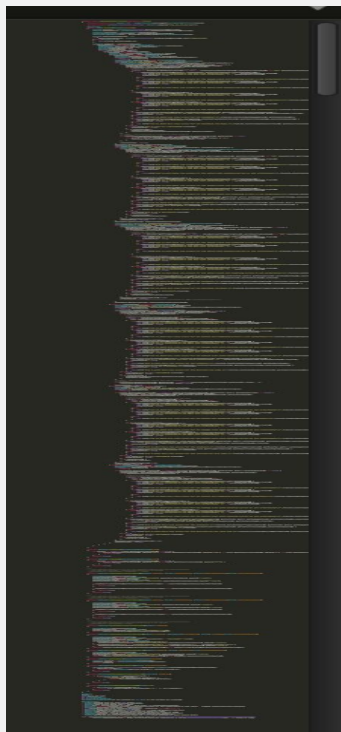
- Multiple classes like this example, one for each record type/source system being returned
- Impossible to unit test or really understand by IT
- Developers added log statements to every branch of code to help “follow” execution path; highly complex
- Upcoming security rules would have added new data types making it worse over time – maintenance nightmare



# RAILINC APPLICATION (RED HAT JBOSS BPM/BRMS)

## Code Quality/Simplicity

Imperative; have to determine what each item is;  
looping; bunches of 'if' statements



```
// If incident is a damage
if (isDamage(railEvent.getNetworkEvent())) {
    if (hasViewAccessForDamage(subject, carMarkOwner, handlingCompanyId, selectedCompanyId)) {
        LOG.info("Uid {},<< Permitted to see DDCT with handling carrier >>" + handlingCompanyId,transId);
        LOG.info("Uid {},<< Permitted to see DDCT with carMarkOwner >>" + carMarkOwner,transId);
        isPermitted = true;
    }else
    {
        LOG.info("Uid {},Dont have view access for the damage event with handling carrier {} and carMarkOwner {}",transId,handlingCompanyId,carMarkOwner);
    }
} else if (isDefect(railEvent.getNetworkEvent())) {
    if (hasViewAccessForDefect(subject, carMarkOwner, handlingCompanyId, selectedCompanyId)) {
        LOG.info("Uid {},<< Permitted to see DDCT with handling carrier >>" + handlingCompanyId,transId);
        LOG.info("Uid {},<< Permitted to see DDCT with carMarkOwner >>" + carMarkOwner,transId);
        isPermitted = true;
    }else
    {
        LOG.info("Uid {},Dont have view access for the defect with handling carrier {} and carMarkOwner {}",transId,handlingCompanyId,carMarkOwner);
    }
}
}
```

Java



```
rule "DDCT - Handling Carrier - User Permission To View Damaged Equipment - Incident - Document"
when
    $document : AssetDocument($id : id,
        industryDocument.documentTypeCode == "DEFECT_CARD",
        $rule : industryDocument.triggeringEvent.industryRuleName in ("107","95","103"),
        $handlingMark : industryDocument.getTriggeringEvent().getHandlingMarkName())
    RailincPrincipals( $selCompany : oneByType(SelectedCompanyPrincipal.class) != null)
    $auth : RailincAuthorizations ( hasAnyRole(((SelectedCompanyPrincipal)$selCompany).companyId, "DDCTSHCBUSR", "DDCTSHCADMIN", "DDCTSHCFUSR", "DDCTSHCROUSR")
        && ((SelectedCompanyPrincipal)$selCompany).companyId == $handlingMark)
then
    $auth.addStringPermission("equipment::assetdocument:DEFECT_CARD:107:" + $id + ":view");
    $auth.addStringPermission("equipment::assetdocument:DEFECT_CARD:95:" + $id + ":view");
    $auth.addStringPermission("equipment::assetdocument:DEFECT_CARD:103:" + $id + ":view");
end
```

Drools

Declarative; say what you want & get it; all looping & 'ifs' statements  
handled by the rules engine

# RAILINC APPLICATION (RED HAT JBOSS BPM/BRMS)

## Code Quality/Simplicity

This block of java code is now used for ALL data types! Literally **turned 1,000's of lines of code into < 10 lines**

```
public RailincResource filterResource(RailincResource railincResource, String typeCode, String subTypeCode) {  
  
    String permissionString = String.format("equipment:*%s:%s:%s:%s:view", railincResource.getResourceType(), typeCode, subTypeCode, railincResource.getResourceId());  
  
    RailincSecurityContext context = RailincSecurityContextHolder.getContext();  
    RailincSubject subject = context.getSubject();  
  
    LOG.debug("filterResource PermissionString is {}", permissionString);  
  
    if (!subject.isPermitted(context.permissionBuilder().resource(railincResource).permissionString(permissionString).build())) {  
        LOG.debug("Loggedin User is {}", subject.getPrimaryPrincipal());  
        return null;  
    }  
  
    return railincResource;  
}
```

Java

# RAILINC APPLICATION (RED HAT JBOSS BPM/BRMS)

## Code Quality/Simplicity

```
rule "DDCT - Handling Carrier - User Permission To View Damaged Equipment Incident - Document"
when
    $document : AssetDocument($id : id,
        industryDocument.documentTypeCode == "DFFECT_CARD",
        $rule : industryDocument.triggeringEvent.industryRuleName in ("107","95","102"),
        $handlingMark : industryDocument.getTriggeringEvent().getHandlingMarkName())

    RailincPrincipals( $selCompany : oneByType(SelectedCompanyPrincipal.class) != null)

    $auth : RailincAuthorizations ( hasAnyRole(((SelectedCompanyPrincipal)$selCompany).companyId, "DDCTSHCBUSR", "DDCTSHCADMIN", "DDCTSHCFUSR", "DDCTSHCRBUSR")
        && ((SelectedCompanyPrincipal)$selCompany).companyId == $handlingMark)
then
```

**Drools**

```
public RailincResource filterResource(RailincResource railincResource, String typeCode, String subTypeCode) {
    String permissionString = String.format("equipment:%s:%s:%s:%s:view", railincResource.getResourceType(), typeCode, subTypeCode, railincResource.getResourceId());

    RailincSecurityContext context = RailincSecurityContextHolder.getContext();
    RailincSubject subject = context.getSubject();

    LOG.debug("filterResource PermissionString is {}", permissionString);

    if (!subject.isPermitted(context.permissionBuilder().resource(railincResource).permissionString(permissionString).build())) {
        LOG.debug("Loggedin User is {}", subject.getPrimaryPrincipal());
        return null;
    }

    return railincResource;
}
```

**Java**

### Benefits

- Easier maintenance
- Faster and easier to research if there's an issue
- Faster and easier to make changes
- Easier if resources move from app to app

# BENEFITS REALIZED IN 2 YEARS

Reduced development costs (greenfield and maintenance)

Faster time to resolve

Increased customer satisfaction

Increased responsiveness to customer needs

Faster on-boarding for new resources and industry committee members

Increased collaboration; Railinc and Industry & Internal Railinc Teams

INTERACT | DISCUSS | DESIGN

# A DESIGN APPROACH TO BRIDGE DEVOPS AND BUSINESS AUTOMATION

Ron Murhammer, NA Business Automation Practice Lead  
David Bush, Business Automation Consultant

Tuesday, May 2, 3:30 PM - 4:15 PM

Located at the Consulting Discovery Zone at the Services Showcase in the Partner Pavilion

To learn more, visit [red.ht/discoverysession](https://red.ht/discoverysession)

RED HAT  
**SUMMIT**

# THANK YOU



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



The logo consists of a red speech bubble shape pointing downwards, containing the text "RED HAT" in a smaller font above "SUMMIT" in a larger, bold font.

**RED HAT**  
**SUMMIT**

**LEARN. NETWORK.  
EXPERIENCE  
OPEN SOURCE.**

# FIELD WORKFORCE MANAGEMENT



- ✓ Resource Planning
- ✓ Work order Scheduling
- ✓ Dispatch workflow
- ✓ Real-time Updates
- ✓ Compliance Rules
- ✓ Work order Completion
- ✓ Signature/Image Capture
- ✓ Reporting and Audit Trail



# MOBILE & BPM FOR FIELD WORKFORCE

