

Not Quite There Yet



- The SkyNet funding bill is passed.
- The system goes online on August 4th, 1997.
- Human decisions are removed from strategic defense.
- SkyNet begins to learn at a geometric rate.
- It becomes self-aware at 2:14am Eastern time, August 29th
- In a panic, they try to pull the plug.
- And, Skynet **fights back**

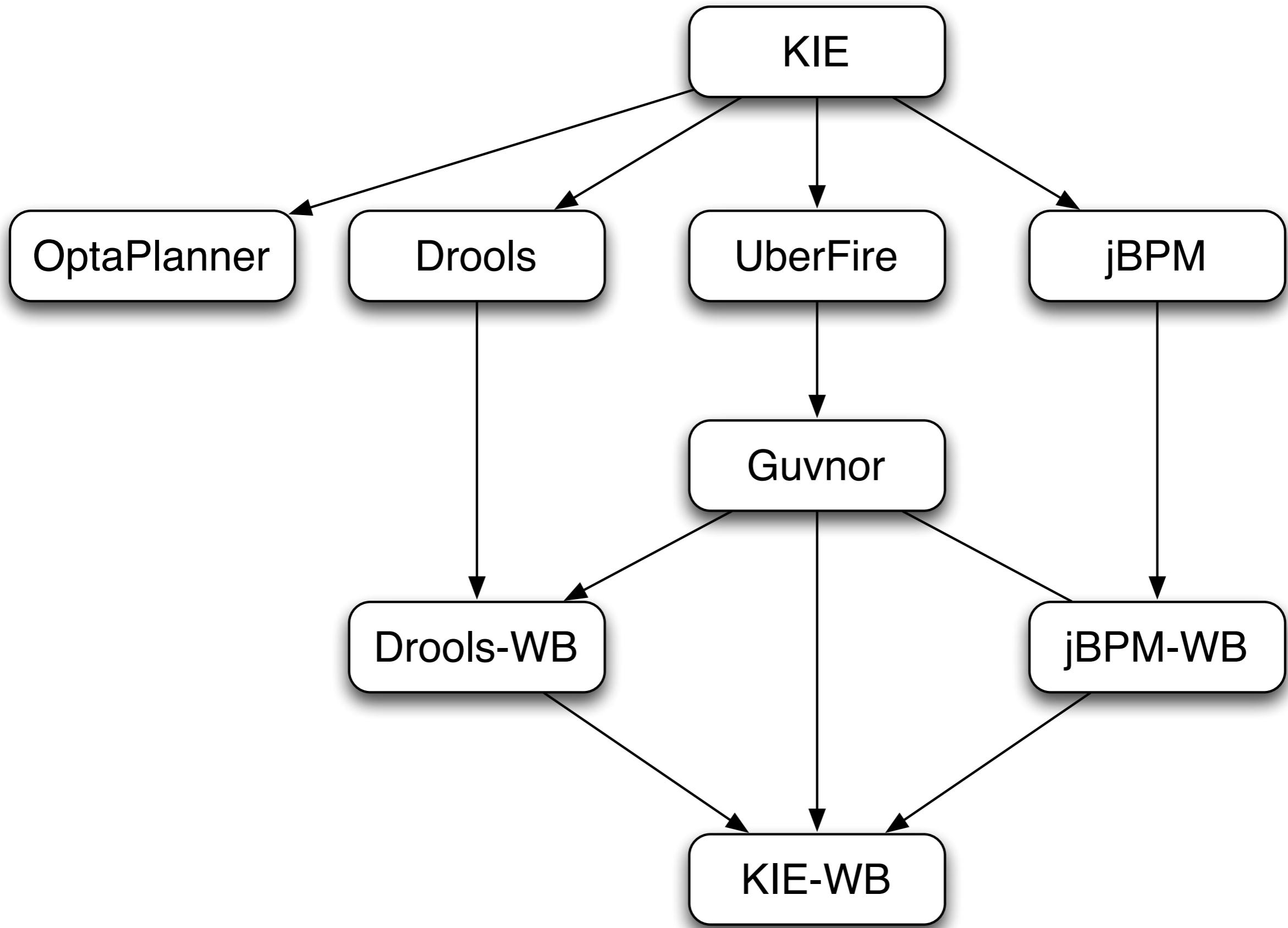
Who am I?



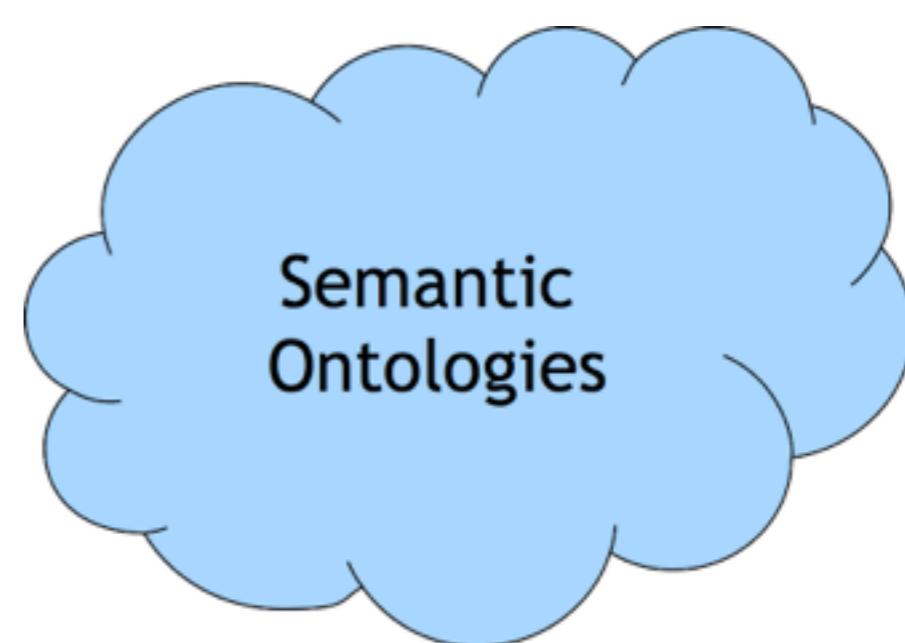
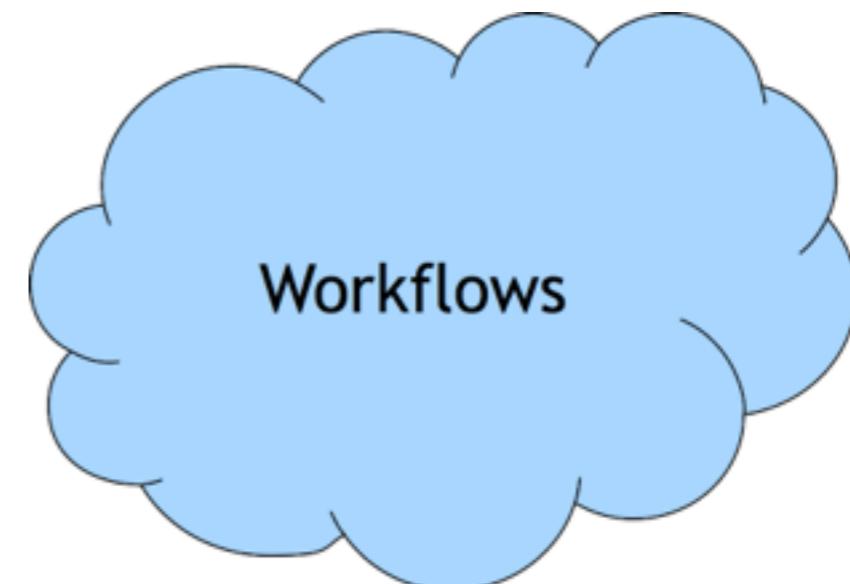
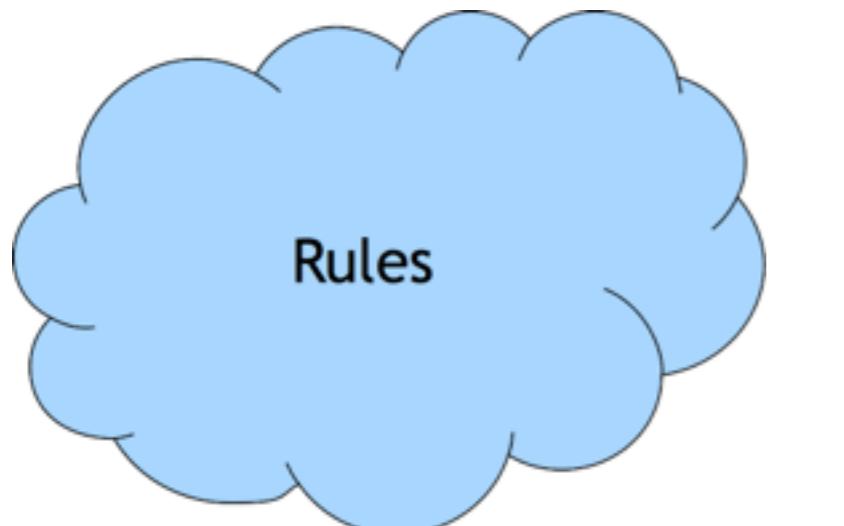
Drools

- Drools co-founder
- JBoss (2005)
- Red Hat (2006)
- Polymita Acquisition 2012
- Red Hat Platform Architect

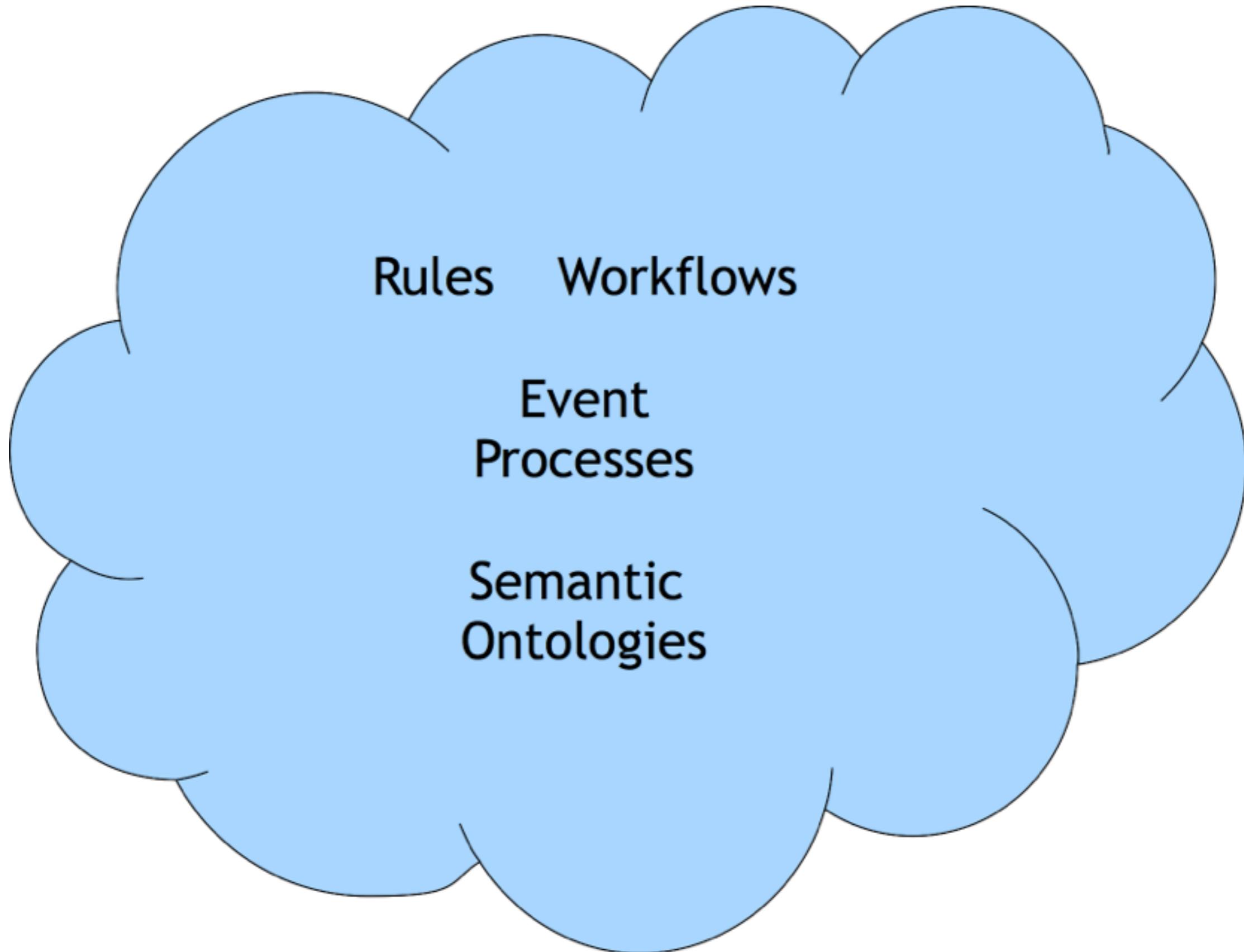
KIE - Knowledge Is Everything



KIE - Knowledge Is Everything



KIE - Knowledge Is Everything

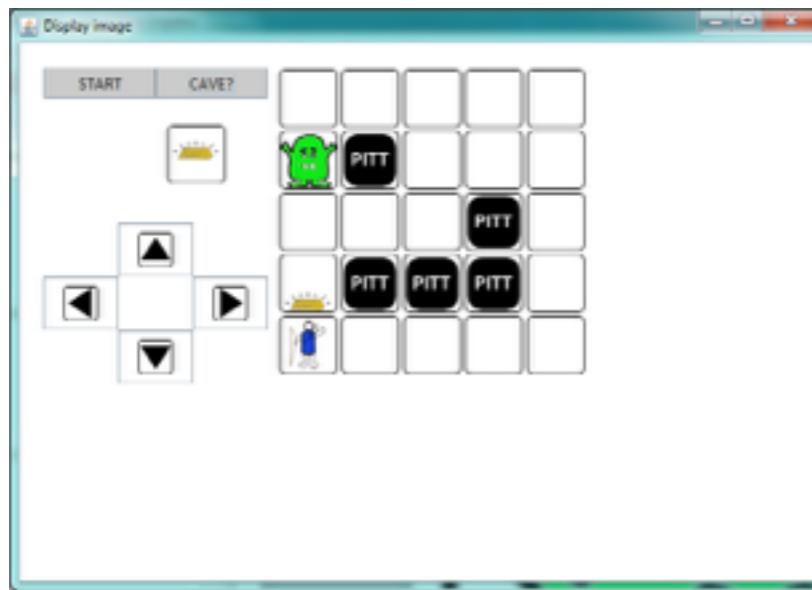


GitHub

- URL:
- <https://github.com/droolsjbpm/>
- Bootstrap project:
- <https://github.com/droolsjbpm/droolsjbpm-build-bootstrap>
- To build all repositories:

```
$ git clone git@github.com:droolsjbpm/droolsjbpm-build-bootstrap.git  
$ droolsjbpm-build-bootstrap/script/git-clone-others.sh  
$ droolsjbpm-build-bootstrap/script/mvn-all.sh clean install -DskipTests
```

Serious Bits :)



<http://www.youtube.com/watch?v=4CvjKqUOEzM>



<http://www.youtube.com/watch?v=Omj4PR3v-nI>



<http://www.youtube.com/watch?v=wORIAZoxttA>

Technical Language

CashFlow Example

Classes

Account	
long	accountNo
int	balance

CashFlow	
Date	date
int	amount

AccountPeriod	
Date	start
Date	end

CashFlow Rule

```
select * from Account acc,
          Cashflow cf, AccountPeriod ap
where acc.accountNo == cf.accountNo and
      cf.type == CREDIT
      cf.date >= ap.start and
      cf.date <= ap.end
acc.balance += cf.amount

rule "increase balance for AccountPeriod Credits"
when
    ap : AccountPeriod()
    acc : Account()
    cf : CashFlow( type == CREDIT,
                   accountNo == acc.accountNo,
                   date >= ap.start && <= ap.end )
then
    acc.balance += cf.amount;
end
```

CashFlow Rule

```
select * from Account acc,  
        Cashflow cf, AccountPeriod ap  
where acc.accountNo == cf.accountNo and  
      cf.type == CREDIT  
      cf.date >= ap.start and  
      cf.date <= ap.end  
acc.balance += cf.amount
```

```
rule "increase balance for AccountPeriod Credits"  
when  
    ap : AccountPeriod()  
    acc : Account()  
    cf : CashFlow( type == CREDIT,  
                  accountNo == acc.accountNo,  
                  date >= ap.start && <= ap.end )  
then  
    acc.balance += cf.amount;  
end
```

CashFlow Rule

```
select * from Account acc,  
        Cashflow cf, AccountPeriod ap  
where acc.accountNo == cf.accountNo and  
      cf.type == CREDIT  
      cf.date >= ap.start and  
      cf.date <= ap.end  
acc.balance += cf.amount
```

```
rule "increase balance for AccountPeriod Credits"  
when  
    ap : AccountPeriod()  
    acc : Account()  
    cf : CashFlow( type == CREDIT,  
                  accountNo == acc.accountNo,  
                  date >= ap.start && <= ap.end )  
then  
    acc.balance += cf.amount;  
end
```

CashFlow Rule

```
select * from Account acc,
          Cashflow cf, AccountPeriod ap
where acc.accountNo == cf.accountNo and
      cf.type == CREDIT
      cf.date >= ap.start and
      cf.date <= ap.end
acc.balance += cf.amount

rule "increase balance for AccountPeriod Credits"
when
    ap : AccountPeriod()
    acc : Account()
    cf : CashFlow( type == CREDIT,
                   accountNo == acc.accountNo,
                   date >= ap.start && <= ap.end )
then
    acc.balance += cf.amount;
end
```

CashFlow Rule

```
select * from Account acc,  
        Cashflow cf, AccountPeriod ap  
where acc.accountNo == cf.accountNo and  
      cf.type == CREDIT  
      cf.date >= ap.start and  
      cf.date <= ap.end  
acc.balance += cf.amount
```

```
rule "increase balance for AccountPeriod Credits"  
when  
    ap : AccountPeriod()  
    acc : Account()  
    cf : CashFlow( type == CREDIT,  
                  accountNo == acc.accountNo,  
                  date >= ap.start && <= ap.end )  
then  
    acc.balance += cf.amount;  
end
```

CashFlow Example

CashFlow			
date	amount	type	accountNo
12-Jan-12	100	CREDIT	1
2-Feb-12	200	DEBIT	1
18-May-12	50	CREDIT	1
9-Mar-12	75	CREDIT	1

AccountingPeriod	
start	end
01-JAN-2012	31-MAR-2012
Account	
accountNo	balance
1	0

rule "Increase balance for AccountPeriod Credits"

when

```
ap : AccountPeriod( )
acnt : Account( )
cf : CashFlow( type == CashFlowType.CREDIT,
               accountNo == acnt.accountNo,
               date >= ap.start && <= ap.end )
```

then

```
modify(acnt) {
    balance = acnt.balance + cf.amount;
}
```

end

rule "Decrease balance for AccountPeriod Debits"

when

```
ap : AccountPeriod( )
acnt : Account( )
cf : CashFlow( type == CashFlowType.DEBIT,
               accountNo == acnt.accountNo,
               date >= ap.start && <= ap.end )
```

then

```
modify(acnt) {
    balance = acnt.balance - cf.amount
}
```

end

CashFlow			
date	amount	type	accountNo
12-Jan-12	100	CREDIT	1
9-Mar-12	75	CREDIT	1

CashFlow			
date	amount	type	accountNo
2-Feb-12	200	DEBIT	1

Account	
accountNo	balance
1	-25

CashFlow Example

```
rule "Print balance for AccountPeriod" salience -50
when
    ap : AccountPeriod()
    acnt : Account( )
then
    System.out.println( "Account Number " + acnt.accountNo +
        " balance " + acnt.balance );
end
```

Agenda		
1	increase balance	arbitrary
2	decrease balance	
3	increase balance	
4	print balance	

CashFlow Example

```
rule "Account in Negative send Warning"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
then
    insert( new NegativeWarning( acnt ) );
end
```

Accumulate

```
rule "accumulate"
when
  acc( b : Bus( color == "red" );
       sumTakings : sum(b.takings); )
then
  System.out.println( "sum is " + sumTakings );
end
```

```
acc( b : Bus( color == "red" );
      sumTakings : sum(b.takings),
      minTakings : min(b.takings),
      maxTakings : max(b.takings);
      minTakings < 100 and maxTakings <
200; )
```

CashFlow Example

```
rule "Account in Negative send Warning"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
then
    insert( new NegativeWarning( acnt ) );
end

rule "Issue Warning"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
    acc( n : NegativeWarning( account == acnt ) ;
        totalWarnings : count(n);
        totalWarnings > 0; )
then
    System.out.println( "Account warning Number " + acnt.accountNo +
        " balance " + acnt.balance + " Warnings " +
        totalWarnings );
    insert( TotalWarnings( acnt, totalWarnings ) );
end
```

Conditional Elements

not Bus (color == “red”)

exists Bus (color == “red”)

forall (Bus (color == “red”)

forall (b : Bus (floors == 2)
 Bus(this == b, color == “red”))

CashFlow Example

```
rule "Issue Warning"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
    acc( n : NegativeWarning( account == acnt ) );
    totalWarnings : count(n);
    totalWarnings > 0; )
then
    System.out.println( "Account warning Number " + acnt.accountNo +
                        " balance " + acnt.balance + " Warnings " totalWarnings );
    insert( TotalWarnings( acnt, totalWarnings ) );
end

rule "Cleanup Warnings"
when
    ap : AccountPeriod()
    acnt : Account( balance >= 0 )
    exists( NegativeWarning( account == acnt ) )
    n : NegativeWarning( account == acnt )
then
    delete( n );
end
```

CashFlow Example

```
rule "Issue Warning"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
    acc( n : NegativeWarning( account == acnt ) over window:time(4w);
        totalWarnings : count(n);
        totalWarnings > 0; )
then
    System.out.println( "Account warning Number " + acnt.accountNo + " balance " +
                        acnt.balance + " Warnings " totalWarnings );
    insertLogical( new TotalWarnings( acnt, totalWarnings ) );
end

rule "Fine"
when
    ap : AccountPeriod()
    acnt : Account( balance < 0 )
    acc( n : NegativeWarning( account == acnt ) over window:time(4w);
        totalWarnings : count(n);
        totalWarnings > 3 )
    TotalWarnings( total < totalWarnings )
then
    System.out.println( "Account warning Number " + acnt.accountNo + " balance " +
                        acnt.balance + " Warnings " totalWarnings );
end
```

Graphical Editors

Data Modeller

Data modeler

Create Save  

Identifier	
Purchase Order	
Purchase Order Header	
Purchase Order Line	

Create new field

*Id Label

*Type

Purchase Order (org.jbpm.examples.purchases.PurchaseOrder)

Position	Identifier	Label	Type	
0	description	Description	String	
1	header	Header	 Purchase Order Header	
2	lines	Lines	 Purchase Order Line [0..N]	
4	requiresCFOApproval		Boolean	
3	total	Total	Double	

Identifier	
description	
Description	
Description	
Type	String
Equals	<input type="checkbox"/>
Position	0

Guided Editors

Guided Editor [Bankruptcy history.rdr]

Save Delete Rename Copy Validate x ▾

EXTENDS None selected ▼

WHEN

1. There is a LoanApplication [a]
The following exists:
There is a Bankruptcy with:
any of the following:
2. yearOfOccurrence --- please choose --- 1990
amountOwed --- please choose --- 10000

THEN

1. Retract LoanApplication [a]
2. Set value of LoanApplication [a] approved false
Set value of LoanApplication [a] explanation has been bankrupt

(show options...)

Decision Table (Extended)

Guided Decision Table [Pricing loans.gdst]

All the rules inherit:None selected

Decision table

Add row... Otherwise Analyze... Audit log

#	Description	amount min	amount max	period	deposit max	income	Loan approved	LMI	rate
1		131000	200000	30	20000	Asset	true	0	2
2		10000	100000	20	2000	Job	true	0	4
3		100001	130000	20	3000	Job	true	10	6

Decision Table (Extended)

Guided Decision Table [Pricing loans.gdst]

All the rules inherit:None selected

Decision table

+ New column
+ Condition columns
+ Action columns
+ (options)

Add row... Otherwise Analyze... Audit log

	#	Description	amount min	amount max	period	deposit max	income	Loan approved	LMI	rate
	#	Description	LoanApplication [application]				IncomeSource	application	application	application
			amount [>]	amount [=≤]	lengthYears [=≡]	deposit [<]	type [=≡]	approved	insuranceCost	approvedRate
+	1		131000	200000	30	20000	Asset	true	0	2
+	2		10000	100000	20	2000	Job			4
+	3		100001	130000		3000			10	6

Decision Table (Square)

Decision table

Add row... Otherwise Analyze... Audit log

#	Description	< 5	≥ 5	Bad	Good	20	25	30
1		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

All the rules inherit:None selected ▾

Rule Templates

Load Template Data

WHEN

There is an Applicant with:

1. age greater than or equal to  
- age less than  



There is a Vehicle with:

2. make equal to  
- model equal to  
- value greater than or equal to  
- value less than  



THEN

Insert Policy:

1. type 
premium 



(show
options...)

Rule Templates

Template Data

Template Data

	minAge	maxAge	make	model	minValue	maxValue	type	premium
	18	25	BMW	318	0	10000	TPFF	1000
	18	25	BMW	318	10000	20000	TPFF	1500
	18	25	BMW	318	0	10000	COMPREHENSIVE	2000
	18	25	BMW	318	10000	20000	COMPREHENSIVE	2500

Template Data

Save and close

Add row...

Decision Table Wizards

Guided Decision Table Wizard X

✓ Summary
✓ Add Fact Patterns
✓ **Add Constraints**
✓ Add Actions to set fields
✓ Add Actions to insert new Facts

Available patterns	Available fields	Conditions
a : Applicant v : Vehicle	this name : Text age : Whole number (integer) [New Predicate]	[minAge] age [maxAge] age

Calculation type: Literal value Formula

Column header (description): maxAge *

Operator: less than

(optional) value list:

Default value:

<- Previous Next -> Cancel Finish

Score Cards

- a) Setup Parameters
- b) Characteristic Section

Scorecard (sc-wge-5)

Setup Parameters

Facts	Resultant Score Field	Initial Score	(a)
Customer	customerScore : double	20	
Use Reason Codes	Resultant Reason Codes Field	Reason Codes Algorithm	Baseline Score
false		none	0.0

Characteristics

New Characteristic

Name	Characteristic	Remove Characteristic	Add Attribute	
Fact		Baseline Score	Reason Code	
Operator	Value	Partial Score	Reason Code	Actions
=	0	10		Remove
>=..<	1,40	20		Remove
>=..<	40,60	25		Remove
>=	60	30		Remove

Score Cards

- UI Generates PMML
- DRL Generated from PMML
- DRL results in
 - Calculated Score
 - Ranked Reason Codes
- Can import PMML 4.1
 - but not exposed yet
- Calculated Scores
 - Currently Summations
 - Weight coming
 - Not in PMML standard

Scorecard (sc-wge-5)

Setup Parameters

Facts	Resultant Score Field	Initial Score
Customer	customerScore : double	20
Use Reason Codes	Resultant Reason Codes Field	Reason Codes Algorithm
false		none
		Baseline Score
		0.0

Characteristics

New Characteristic

Name	Characteristic	Remove Characteristic	Add Attribute	
CustAgeScore	customerAge : int	Baseline Score	Reason Code	
Customer				
Operator	Value	Partial Score	Reason Code	Actions
=	0	10		Remove
>=,<	1,40	20		Remove
>=,<	40,60	25		Remove
>=	60	30		Remove

Name	Characteristic	Remove Characteristic	Add Attribute	
AppLicenceScore	validLicense : boolean	Baseline Score	Reason Code	
Applicant				
Operator	Value	Partial Score	Reason Code	Actions
true	N/A	5		Remove
false	N/A	-5		Remove

Name	Characteristic	Remove Characteristic	Add Attribute	
CustResScore	placeOfResidence : String	Baseline Score	Reason Code	
Customer				
Operator	Value	Partial Score	Reason Code	Actions
=	USA	15		Remove
in	FRANCE,UK,SPAIN	17		Remove
=	INDIA	19		Remove

Scenarios

Run scenario

+ GIVEN

insert [Applicant][app]

age: 20

insert [Vehicle][v]

make: BMW

model: 318

value: 15000

+ CALL METHOD

Add input data and expectations here.

+ EXPECT

⌚ Use real date and time

A fact of type [Policy] has values:

type: equals TPFT

premium: equals 1500

A fact of type [Policy] has values:

type: equals COMPREHENSIVE

premium: equals 2500

More...

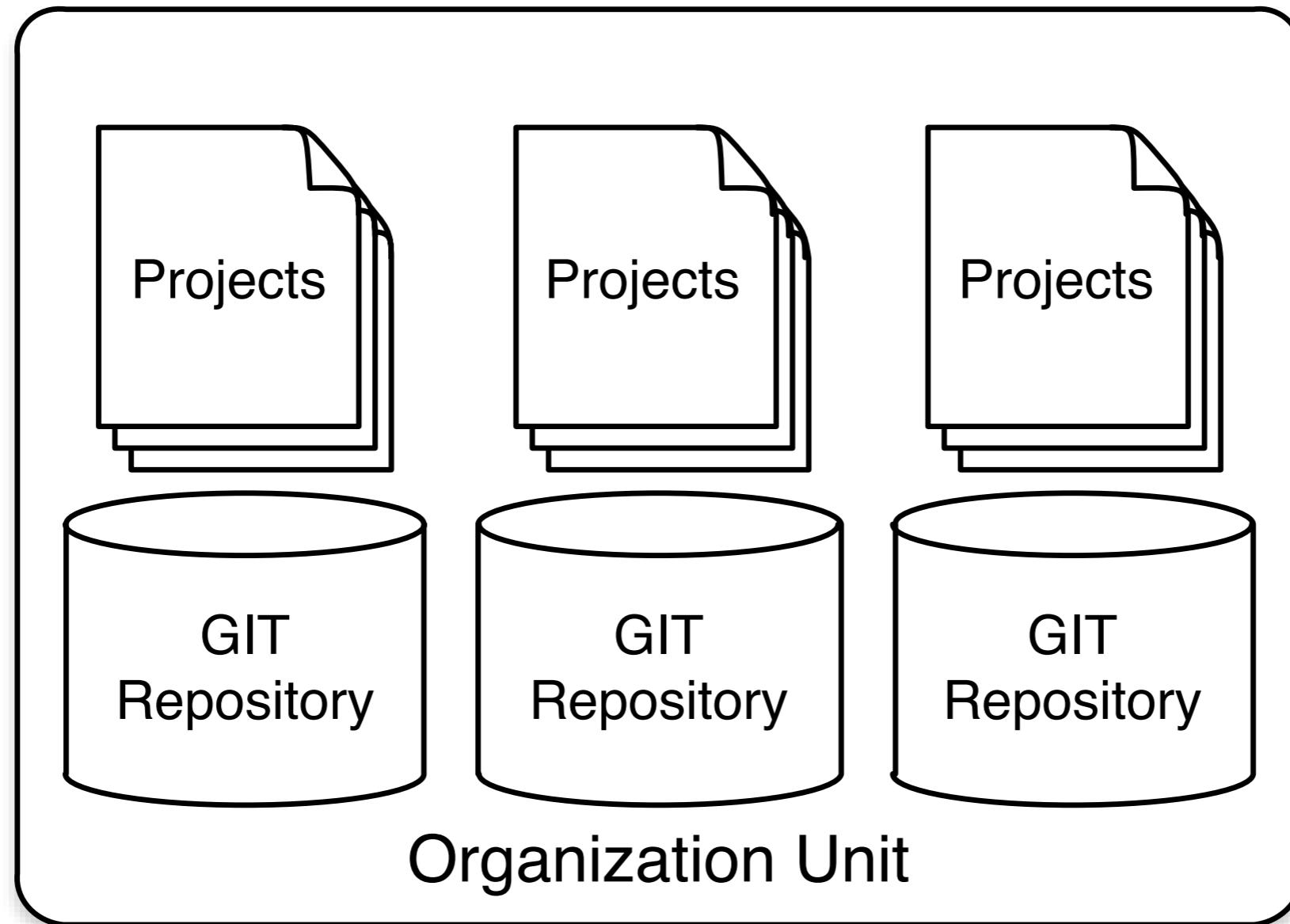
(configuration)

+ (globals)

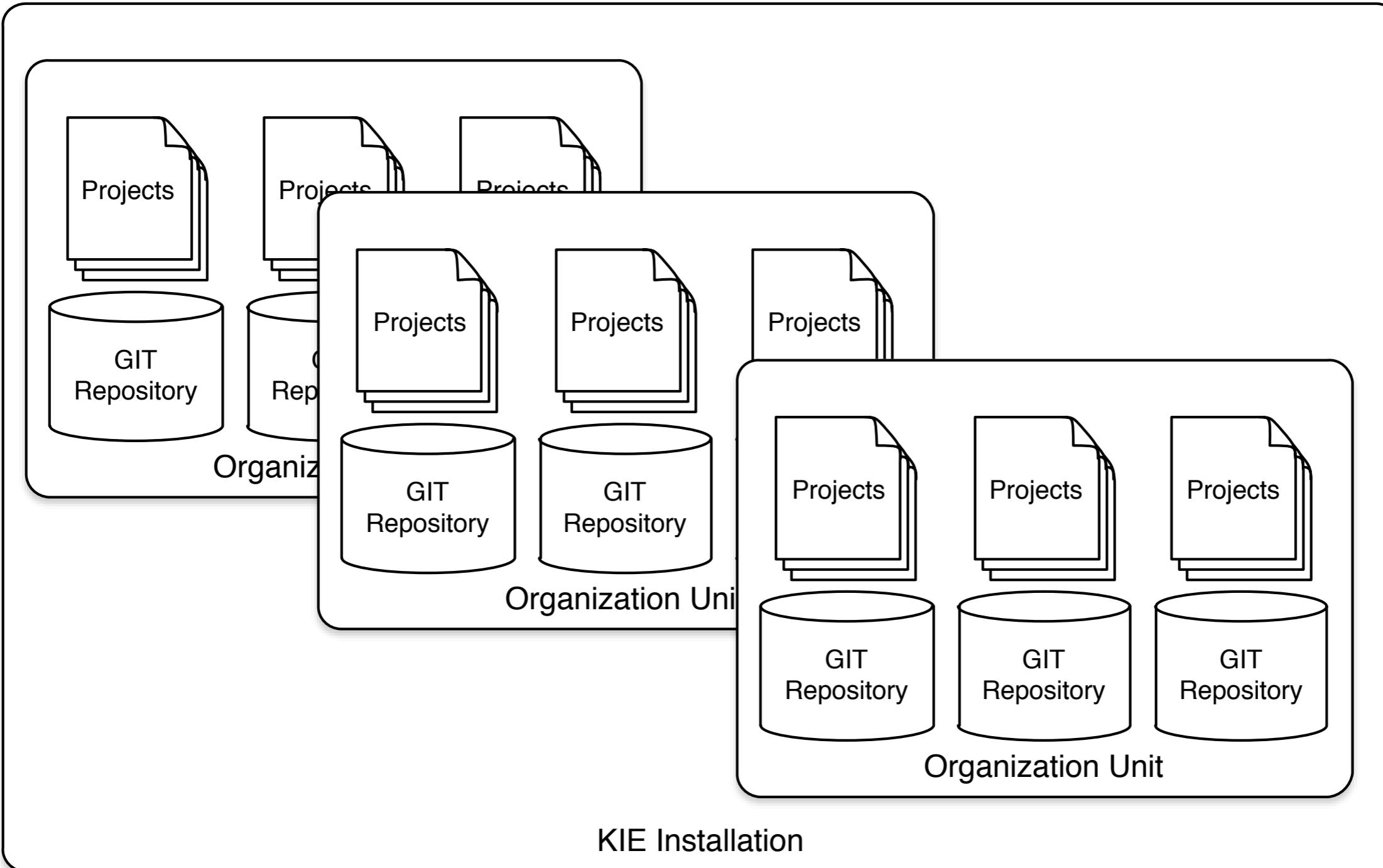
All rules may fire

Architecture

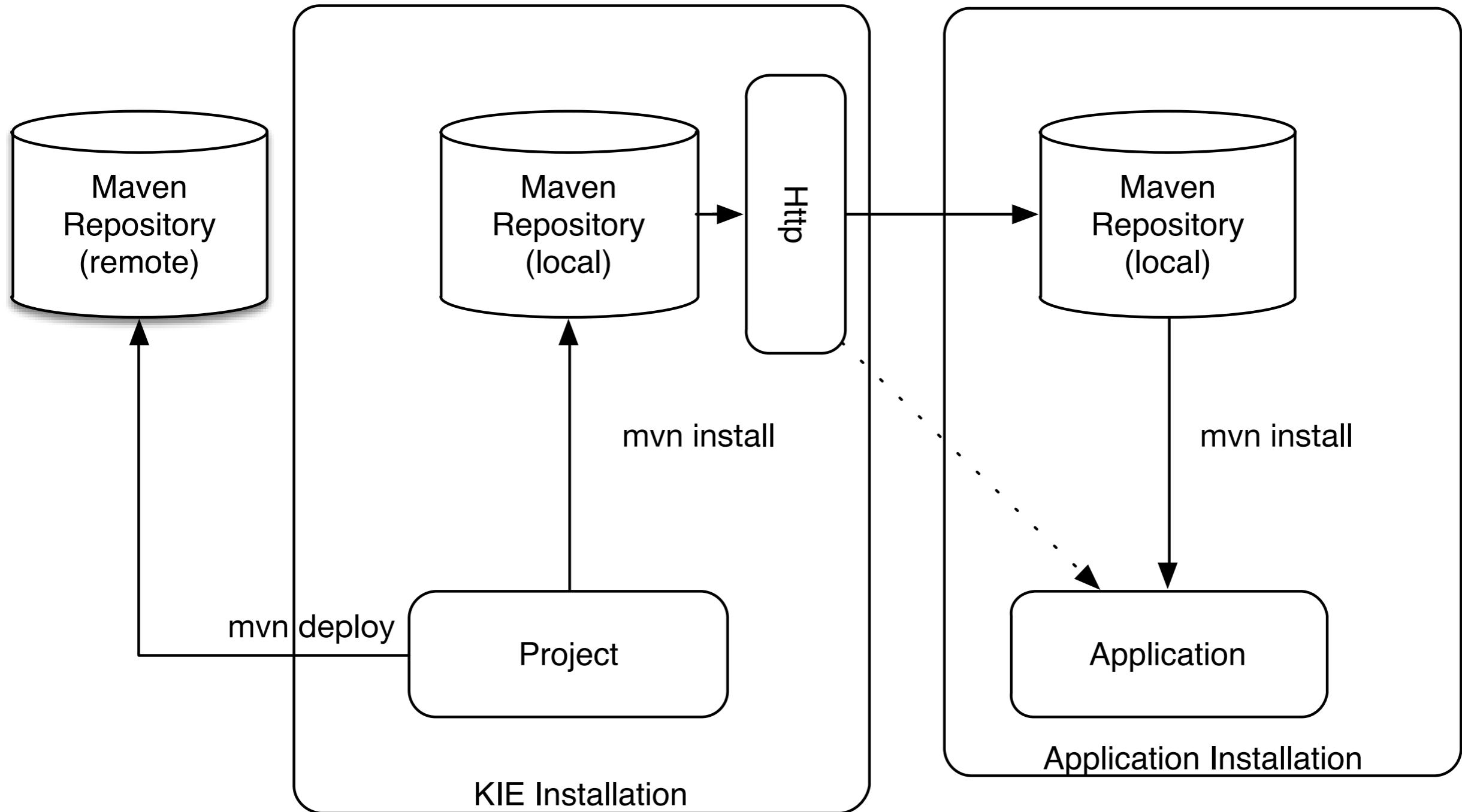
GIT + Maven



KIE Installation



Application Architecture



Workbench

5.x Critique

Welcome: admin [Sign Out]

Drools

Navigate

Find br.com.redhat.sugar Business rule asset AccountManager

Browse Save changes Save and close Select Working Sets Validate Verify View source Actions... Status: [Draft]

Knowledge Bases

Create New

- Packages
 - br
 - com
 - redhat
 - sugar
- Business rule asset
- Technical rule asset
- (x)= Functions
- DSL configurations
- Model
- Processes
- Enumerations
- Test Scenarios
- XML, Properties
- Other assets, docur
- WorkingSets

defaultPackage

QA

Package snapshots

Administration

WHEN

There is an Operation [op] with:

1. lead --- please choose --- doNotCall equal to

1. Modify value of Operation [op] role ACCOUNTMANAGE

THEN

(options)

Attributes:

no-loop

dialect java

AccountManagerLeads

Title: AccountManagerLeads

Categories: CRM/Sugar

Modified Fri 05 Nov 2010 04:35:15
on: PM BRT
by: admin
Note:
Created Thu 04 Nov 2010 04:31:15
on: PM BRT
Created admin
by:
Format: brl
Package: br.com.redhat.sugar
Is
Disabled:
UUID: 2b4582db-184c-461c-aa3d-e2e320f1a038
Other meta data ...
Version history ...

Description:

Close all items

Done

YSlow

5.x Critique

JCR

- Performance Issues
- Everything stored as blob
- No tagging, branching etc.
- Webdav
- Limited team providers

Deployment

- Binary blobs, on url
- Simple Snapshot system
- No real methodology
- Doesn't align with any industry standards

Content

- Single tree structure (packages)
- Packages created project deployment units
- No easy rule use
- Loading “model” jars into packages

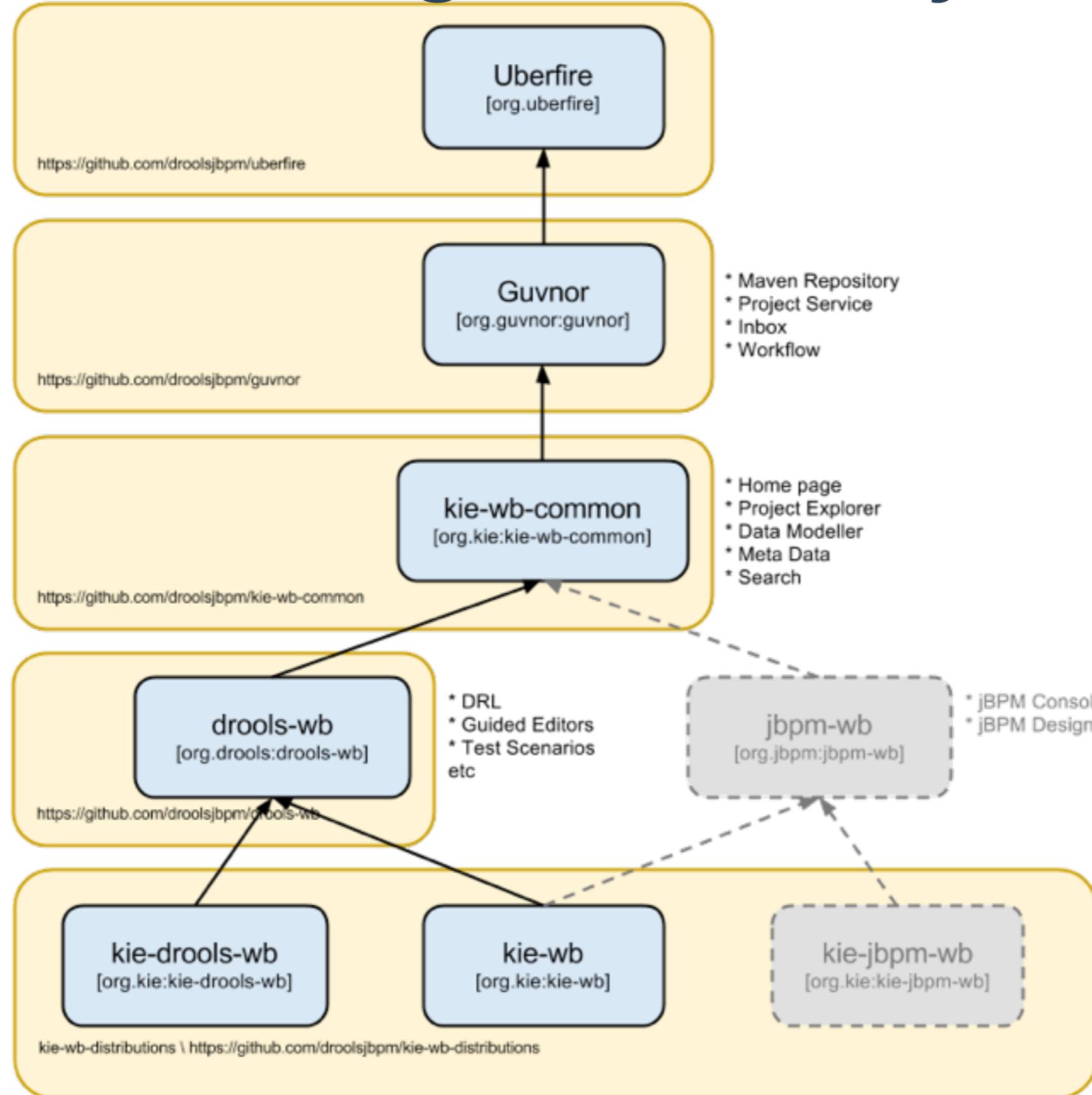
UI

- Not easily extended
- Fixed layouts
- No perspectives
- Hard to change
- Too many hacks

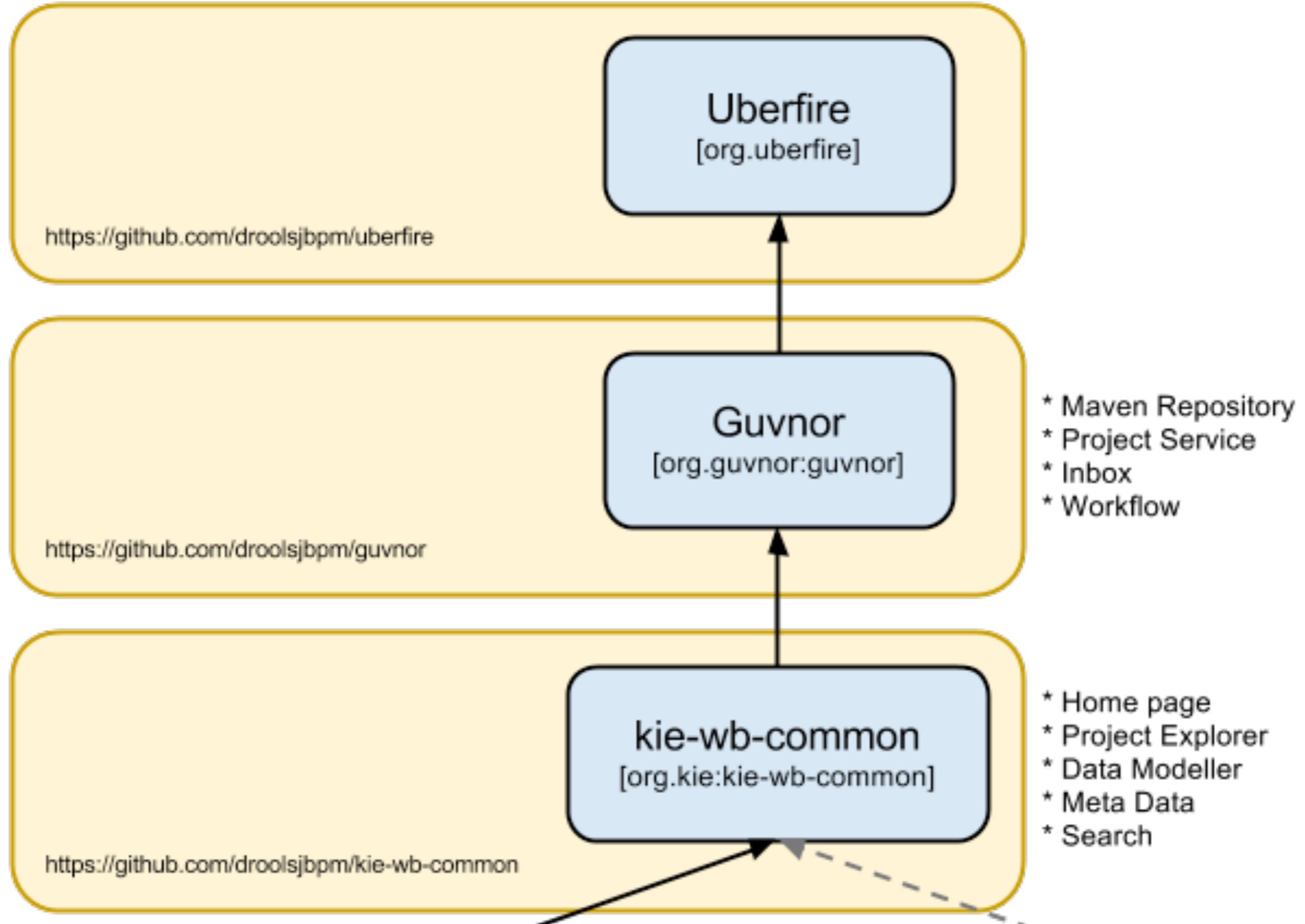




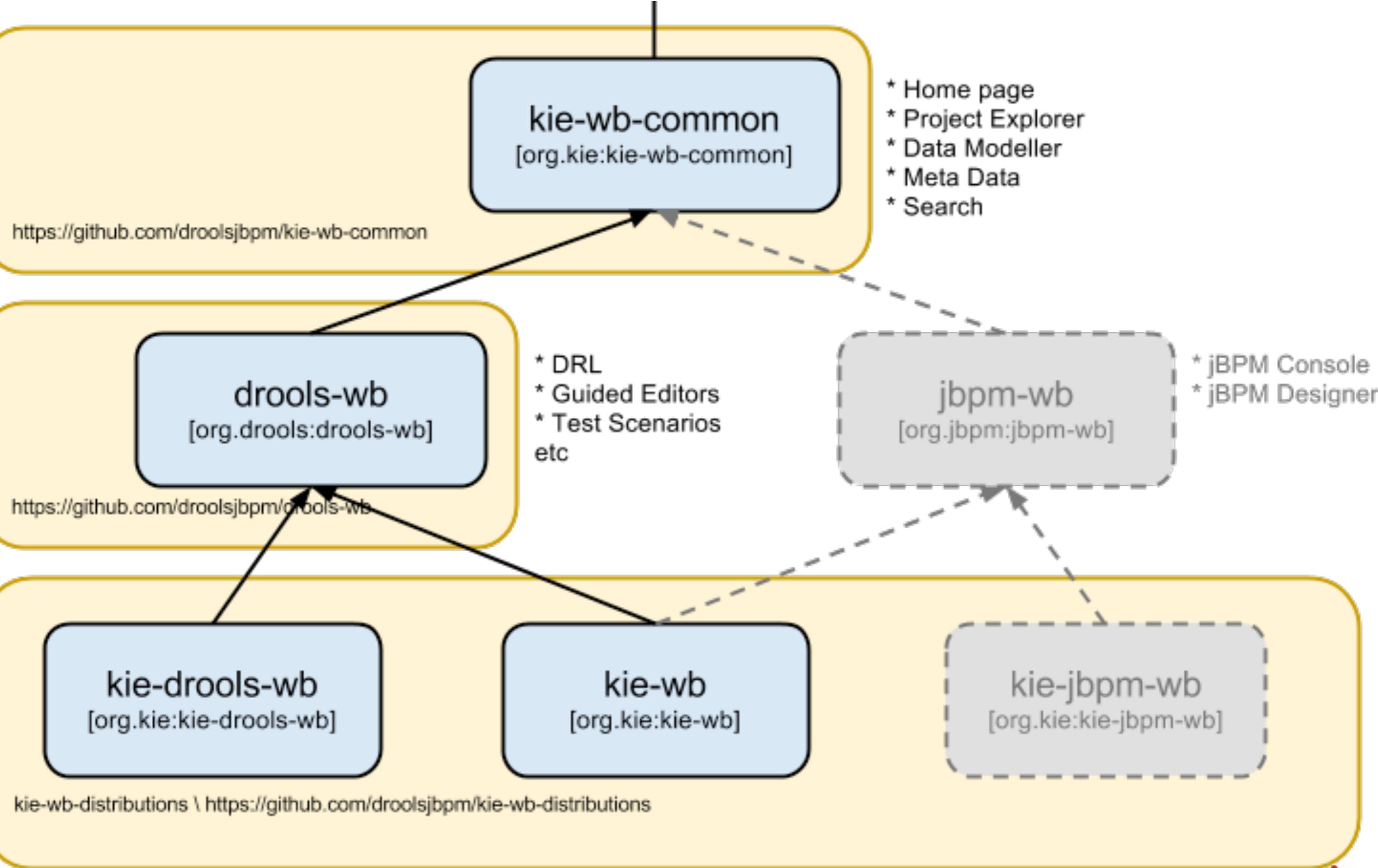
KIE - Knowledge Is Everything



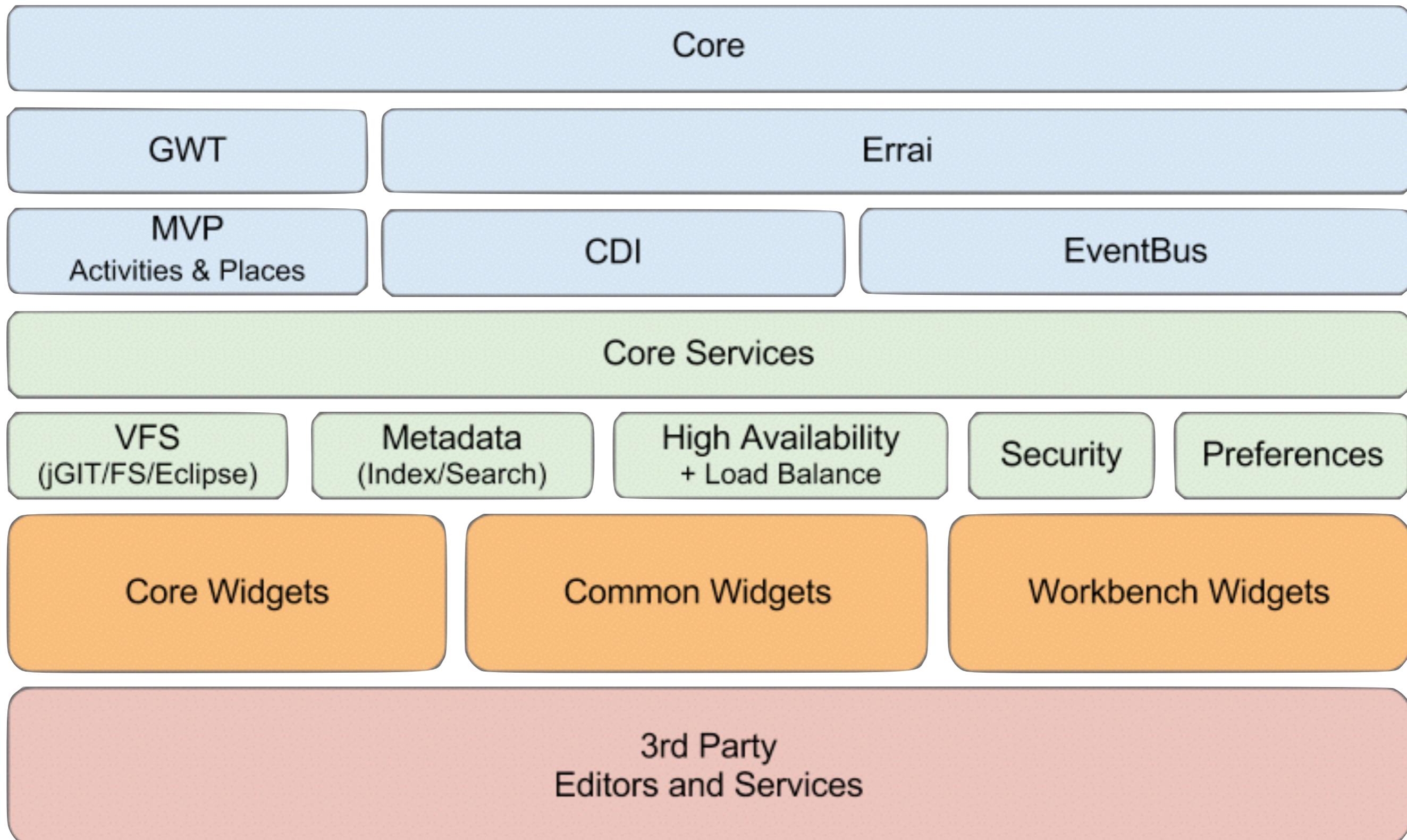
KIE - Knowledge Is Everything



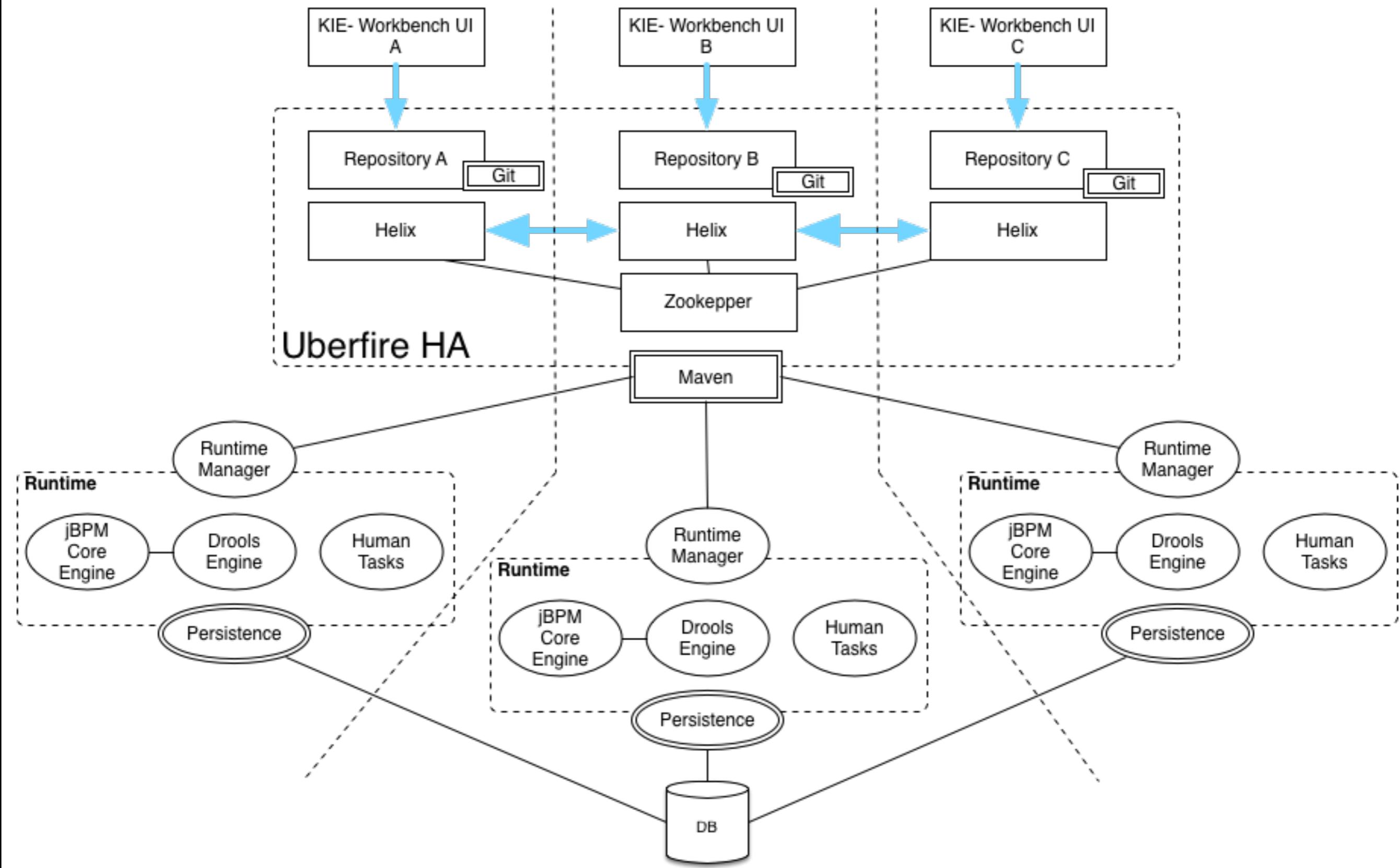
KIE - Knowledge Is Everything



UberFire Architecture Overview



High Availability



Workbench Screens

The screenshot shows a workbench interface with the following components:

- Top Bar:** Home, Perspectives, Logout, Explore, New, Tools, Find, Search...
- Left Sidebar:** Administration, Project Explorer, Guvnor M2 Repository Explorer.
- Project Explorer:** Shows a tree view for 'uf-playground.git / mortgages /' containing: .., Java, Resources, Tests (Java), Tests (Resources), Project definition 'mortgages', and External imports. The 'External imports' node is highlighted with a red box.
- Central Area:** A 'Decision table' editor titled 'Pricing loans.gdst'. It has tabs: Add row..., Otherwise, Analyze..., Audit log. The 'Otherwise' tab is selected. The table has columns: #, Description, amount min, amount max, period, deposit max, income, and Loan approv. Data rows:

#	Description	amount min	amount max	period	deposit max	income	Loan approv
1		131000	200000	30	20000	Asset	true
2		10000	100000	20	2000	Job	true
3		100001	130000	20	3000	Job	true
- Bottom Left:** Recently Edited, Refresh, Open selected, Delete selected. A table showing recently edited files: Format, Name, Date, Open. One entry: Dummy rule.drl, May 22, 2013, Open button.
- Bottom Right:** Problems view. Table headers: Level, Text, File, Artifact ID, Column, Line. One entry: [ERR 102] Line 7:0 mismatched input 'then' in Dummy rule.drl org.test.mortgages 0 7.

Workbench Screens

- Workbench Screen
 - DIV
 - Lifecycle events
 - OnStartUp, OnShutDown
 - OnOpen, OnMayClose, OnClose
 - OnFocus, OnLostFocus

Workbench Screens

```
@WorkbenchScreen(identifier = "MyFirstPanel")
public class MyFirstPanel extends SimplePanel {

    public MyFirstPanel() {
        setWidget( new Label("Hello World 1") );
    }

}

@WorkbenchPartTitle
public String myTitle() {
    return "My First Panel!";
}

}
```

Workbench Editor

Home Perspectives ▾ Logout Explore ▾ New ▾ Tools ▾ Find Search...

Project Explorer Guvnor M2 Repository Explorer

uf-playground.git / mortgages /

- ..
- Java
- Resources
- Tests (Java)
- Tests (Resources)
- Project definition 'mortgages'
- External imports

Recently Edited

Refresh Open selected Delete selected

Format	Name	Created	Open
rule.drl	Dummy rule.drl	2013 May 22 05:50:33	<button>Open</button>

ed Decision Table [Pricing loans.gdst] ×

Decision table

Add row... Otherwise Analyze... Audit log

#	Description	amount min	amount max	period	deposit max	income	Loan appro
1		131000	200000	30	20000	Asset	true
2		10000	100000	20	2000	Job	true
3		100001	130000	20	3000	Job	true

Edit Source Config Metadata

Problems

Level	Text	File	Artifact ID	Column	Line
[ERR 102]	[ERR 102] Line 7:0 mismatched input 'then' in	Dummy rule.drl	org.test.mortgages	0	7

JBoss® by Red Hat

Workbench Editor

- Workbench Screen
 - DIV
 - Lifecycle events
 - OnStartUp, OnShutDown
 - OnOpen, OnMayClose, OnClose
 - OnFocus, OnLostFocus
 - **IsDirty, OnSave**

Life Cycle Annotation

```
@WorkbenchEditor(identifier = "TextEditor",
    supportedTypes = { TextResourceType.class,
                        DotResourceType.class } )
public class TextEditorPresenter {
    (...)

    @OnStart
    public void onStart( final Path path ) {
        this.path = path;
    }

    @OnSave
    public void onSave() {
    }

    @IsDirty
    public boolean isDirty() {
        return view.isDirty();
    }
}
```

Life Cycle Annotation

```
@WorkbenchEditor(identifier = "TextEditor",
    supportedTypes = { TextResourceType.class,
                        DotResourceType.class } )
public class TextEditorPresenter {
    (...)

    @OnStart
    public void onStart( final Path path ) {
        this.path = path;
    }

    @OnSave
    public void onSave() {
    }

    @IsDirty
    public boolean isDirty() {
        return view.isDirty();
    }
}
```

Workbench Perspective

The screenshot shows the JBoss Geronimo Workbench Perspective. At the top, there is a navigation bar with links for Home, Perspectives (which is highlighted with a red box), Logout, Explore, New, Tools, Find, and a search bar. Below the navigation bar, there are several tabs: Administration, Author, Project Explorer, and Guvnor M2 Repository Explorer. The 'Guvnor M2 Repository Explorer' tab is currently active, displaying a 'Decision table' editor. The table has columns for #, Description, amount min, amount max, period, deposit max, income, and Loan approv. There are three rows with data: Row 1 (amount min: 131000, amount max: 200000, period: 30, deposit max: 20000, income: Asset, Loan approv: true); Row 2 (amount min: 10000, amount max: 100000, period: 20, deposit max: 2000, income: Job, Loan approv: true); and Row 3 (amount min: 100001, amount max: 130000, period: 20, deposit max: 3000, income: Job, Loan approv: true). To the left of the table, there is a 'Project Explorer' view showing the project structure: 'uf-playground.git / mortgages /'. The structure includes '..', 'Java', 'Resources', 'Tests (Java)', 'Tests (Resources)', 'Project definition 'mortgages'', and 'External imports'. Below the table, there are tabs for Edit, Source, Config, and Metadata. A 'Problems' view is also visible at the bottom, showing a single error: [ERR 102] Line 7:0 mismatched input 'then' in Dummy rule.drl org.test.mortgages 0. On the far left, there is a 'Recently Edited' list with 'Refresh', 'Open selected', and 'Delete selected' buttons. At the bottom left, there is a table showing recently edited files: Name, Created, Format, Date, and Open. The first file listed is 'rule.drl' (Format: Dummy, Created: 2013 May 22 05:50:33, Open button). A 'JBoss by Red Hat' logo is located in the bottom right corner.

Format	Name	Created	Date	Open
Dummy	rule.drl	2013 May 22 05:50:33		<button>Open</button>

Level	Text	File	Artifact ID	Column	Line
[ERR 102]	Line 7:0 mismatched input 'then' in	Dummy rule.drl	org.test.mortgages	0	7

Workbench Perspective

```
$registerPerspective({
  "id": "Markdown Editor",
  "view": {
    "parts": [
      {
        "place": "MarkdownLiveViewer",
        "parameters": {}
      }
    ],
    "panels": [
      {
        "width": 600,
        "min_width": 300,
        "position": "west",
        "parts": [
          {
            "place": "MarkdownLiveEditor",
            "parameters": {}
          }
        ]
      }
    ]
  },
  on_close: function () {
  }
});
```



Workbench Home Page

KIE Workbench

Home Authoring Deploy Process Management Tasks Dashboards Find User: salaboy ?

Search...

The Knowledge Life Cycle

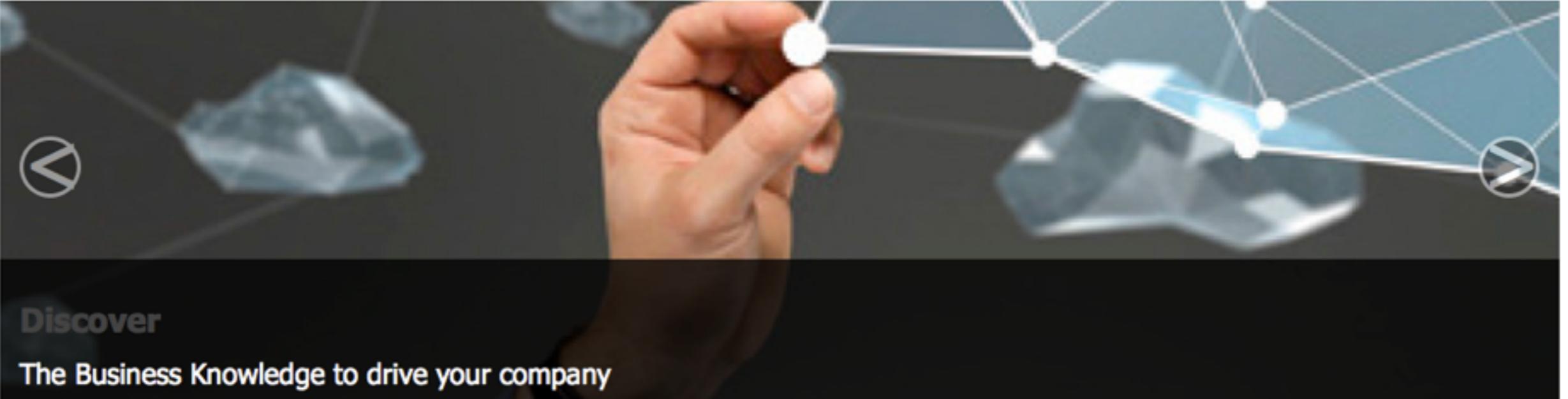
Authoring Deploy Process Management Tasks Dashboards

Project Authoring Deployments Process Definitions Tasks List

Artifact repository Jobs Process Instances

Administration

Business Dashboards



Discover
The Business Knowledge to drive your company

Helpful InfoPops

Help

The screenshot shows a modularized workbench interface. On the left, a sidebar titled "Project Explorer" lists various assets like "Mortgage", "Guided Decision Table", "Decision Rule", "Decision Rule Set", "Decision Model", and "Decision Model Set". A callout box labeled "Project Explorer" points to this sidebar. In the center, there's a "Content Area" where "Opened assets are added here". A callout box labeled "Content Area" points to this area. At the bottom, a "Problems" section displays validation errors for the current project. A callout box labeled "Problems" points to this section.

Project Explorer
Use the project explorer to navigate projects for your organizational's repository and create assets to author.

Content Area
Opened assets are added here

Problems
When you save, validation errors for the current project are shown here

Authoring
Modularized and customizable workbench

Don't show again

Close



KIE Workbench

Home Authoring Deploy Process Management Tasks Dashboards

Find User: jbpm

Explore New Item Tools

search...



Project Explorer

example / uf-playground / mortgages

 <default>
 org
 mortgages

DRL

DOMAIN SPECIFIC LANGUAGE DEFINITION

ENUMERATION DEFINITION

GUIDED DECISION TABLE

GUIDED RULE

Bankruptcy history

No bad credit checks

no NINJAs

Underage

GUIDED RULE (WITH DSL)

JAVA SOURCE FILES

Guided Editor [Bankruptcy history]

EXTENDS

None selected

WHEN

1. There is a LoanApplication [a]

The following exists:

There is a Bankruptcy with:
any of the following:

2. yearOfOccurrence greater than

1990

- amountOwed greater than

10000

THEN

1. Retract LoanApplication [a]

Set value of LoanApplication [a]

approved

false

2. Set value of LoanApplication [a]

explanation

has been bankrupt

(show options...)

Edit Source Config Metadata

Problems

Level	Text	File	Column	Line
-------	------	------	--------	------

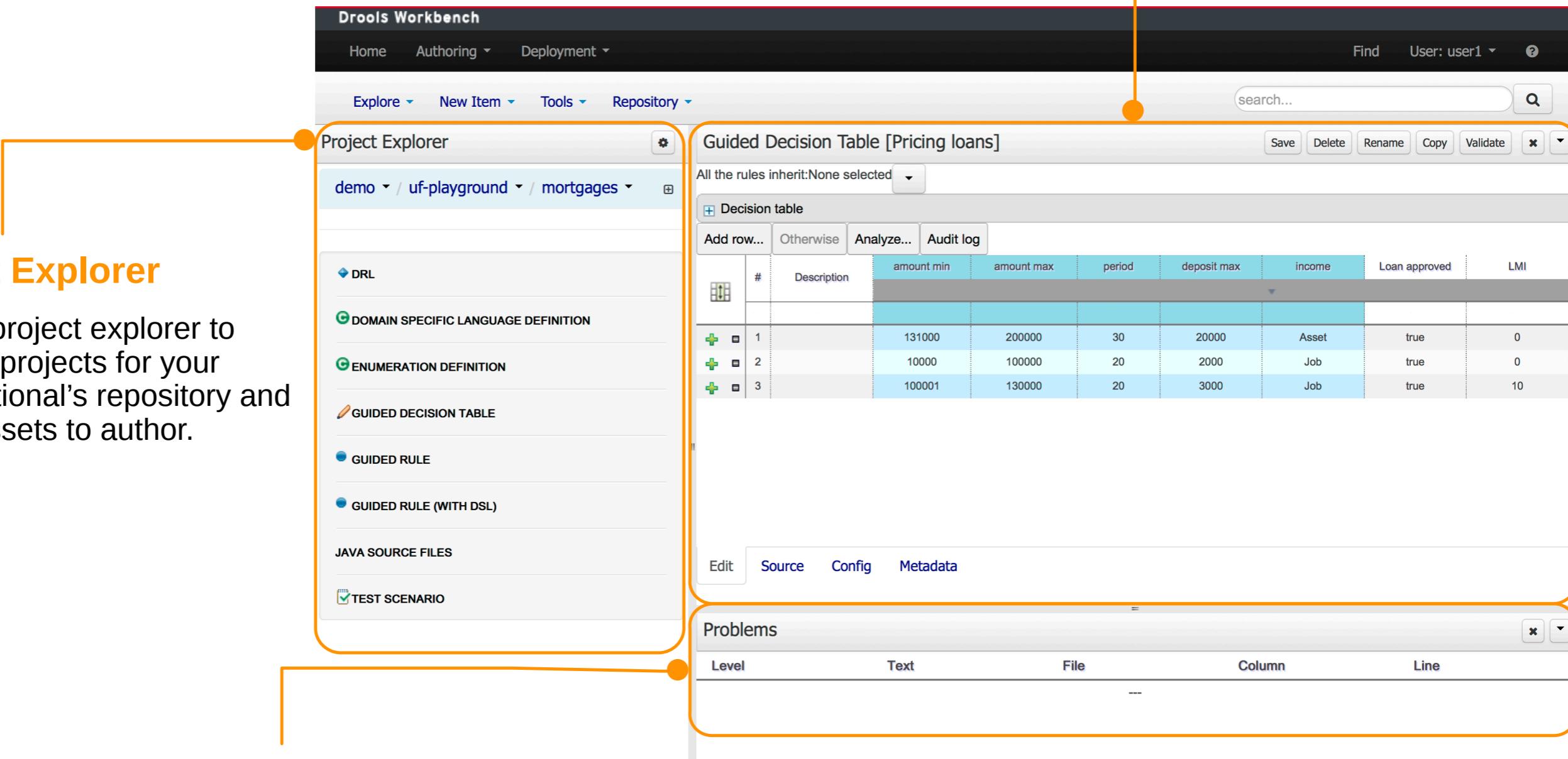
Authoring Perspective

Project Explorer

Use the project explorer to navigate projects for your organizational's repository and create assets to author.

Content Area

Opened assets are added here



Problems

When you save, validation errors for the current project are shown here

Tab Stack

Asset List

The opened assets are shown here. Click to continue authoring.

Guided Editor [CreditApproval]

- E Guided Editor [CreditApproval]
- V Guided Decision Table [Pricing loans]
- Test Scenario [Are they old enough]

THEN

1. Approve the loan

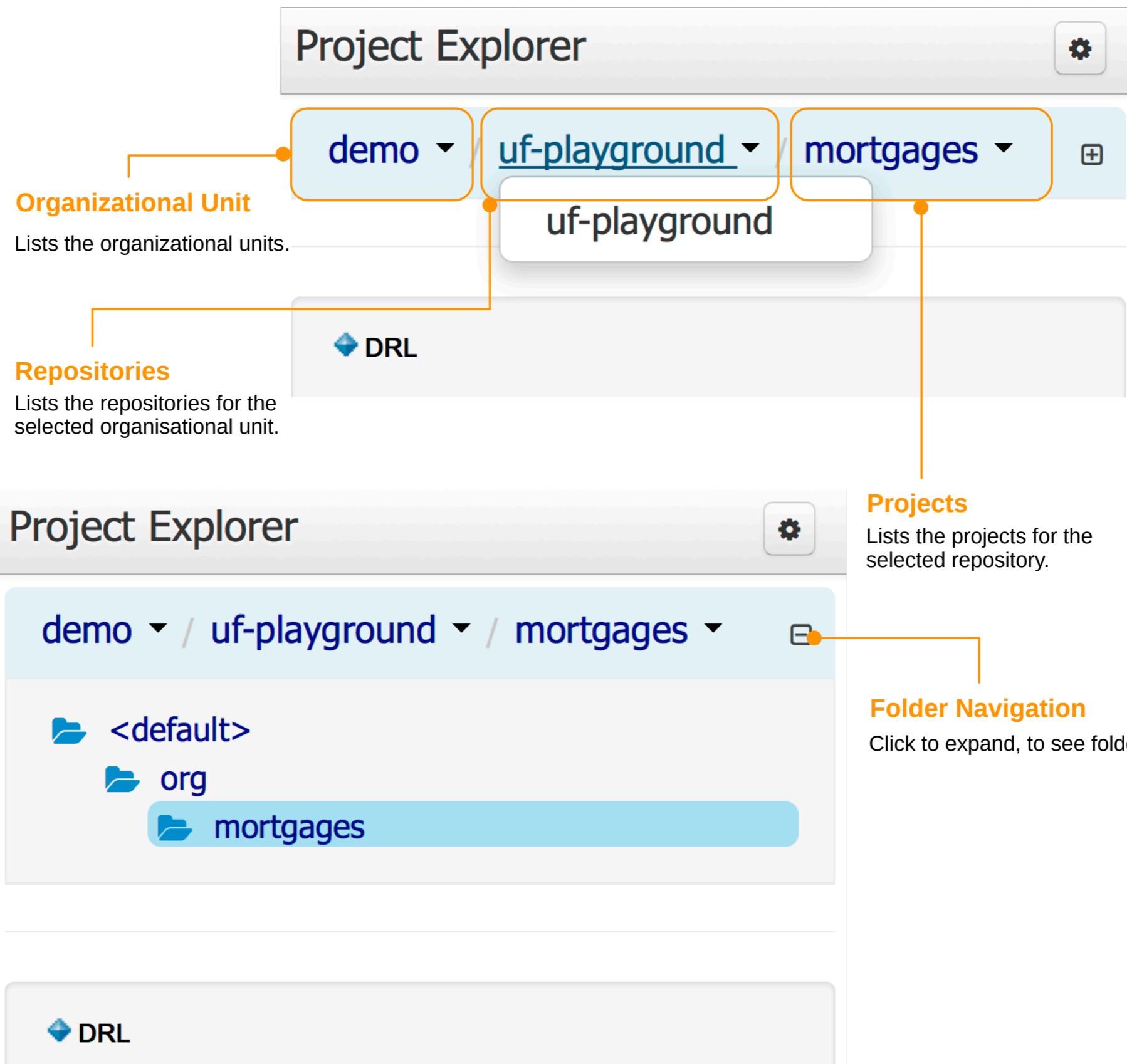
(show options...)

Tab Stack Button

Click the button to see the list of opened assets



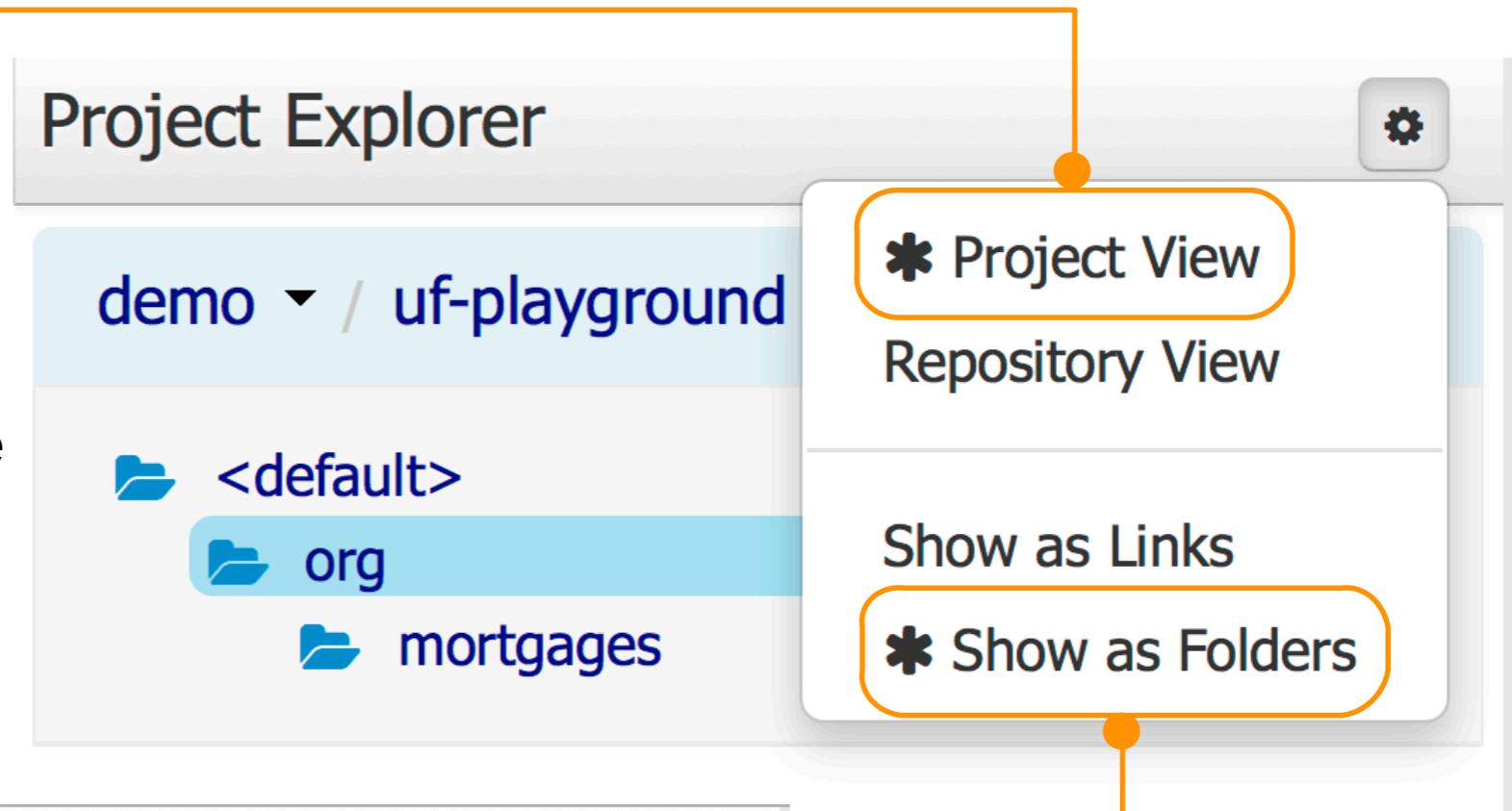
Project Explorer



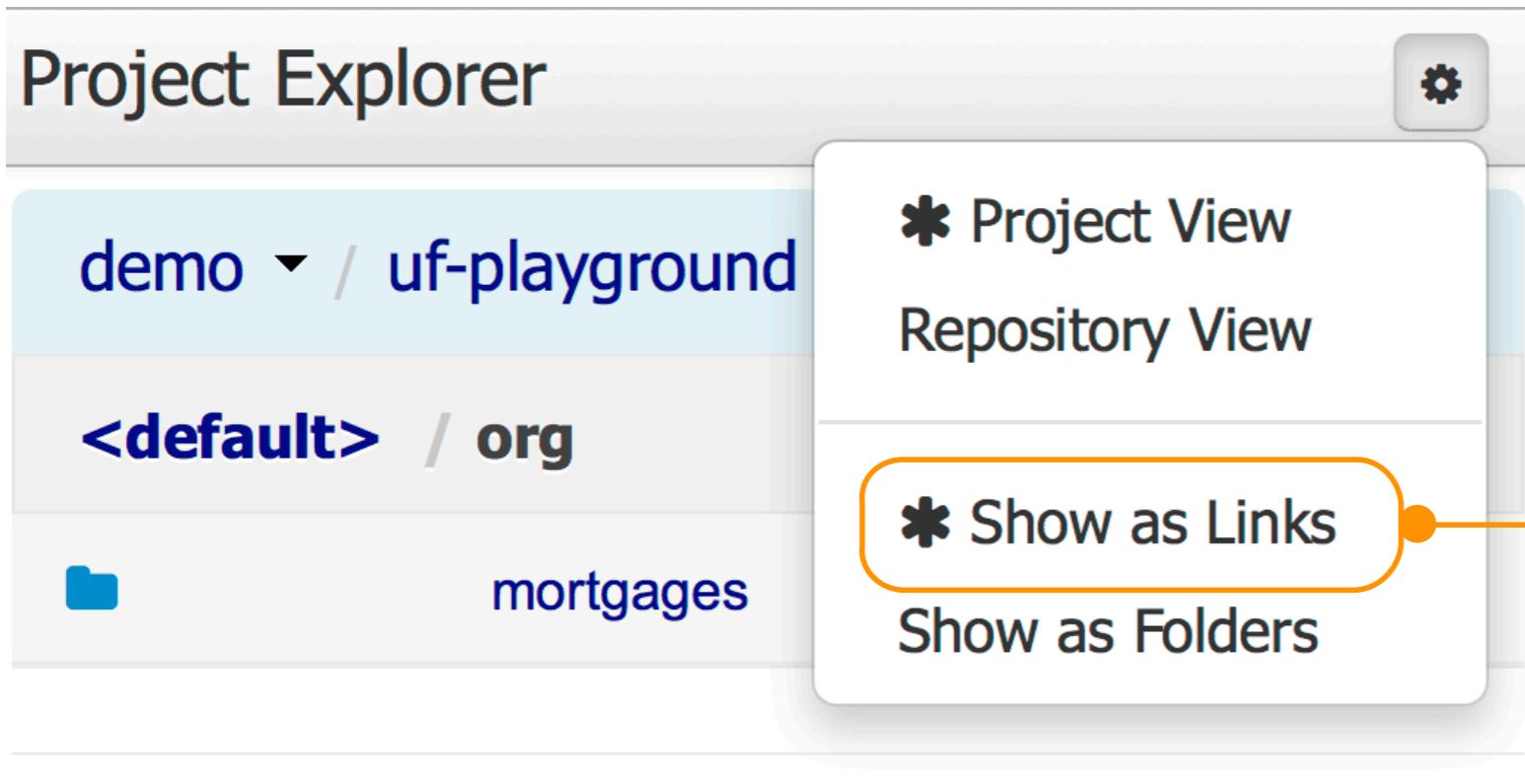
Project View

Project View

The project view is a simplified and hides many technical aspects of the project. Such as merging multiple paths into one tree.



Project Explorer



Show as Folders

Shows the project folders using a tree view.

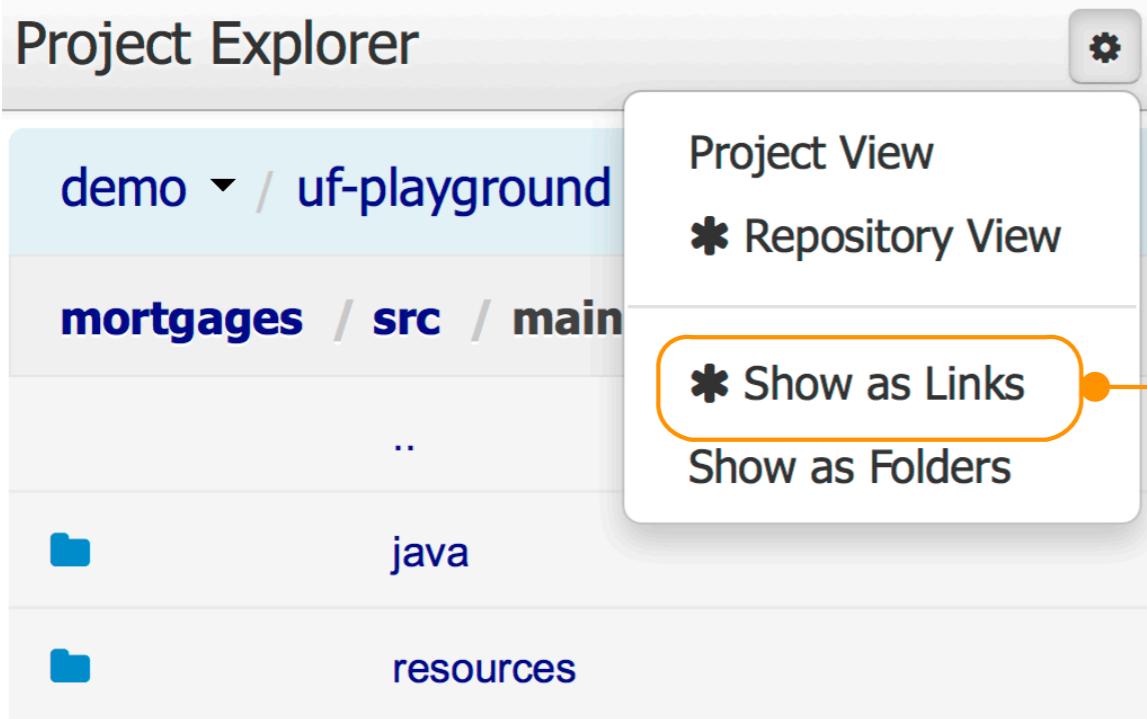
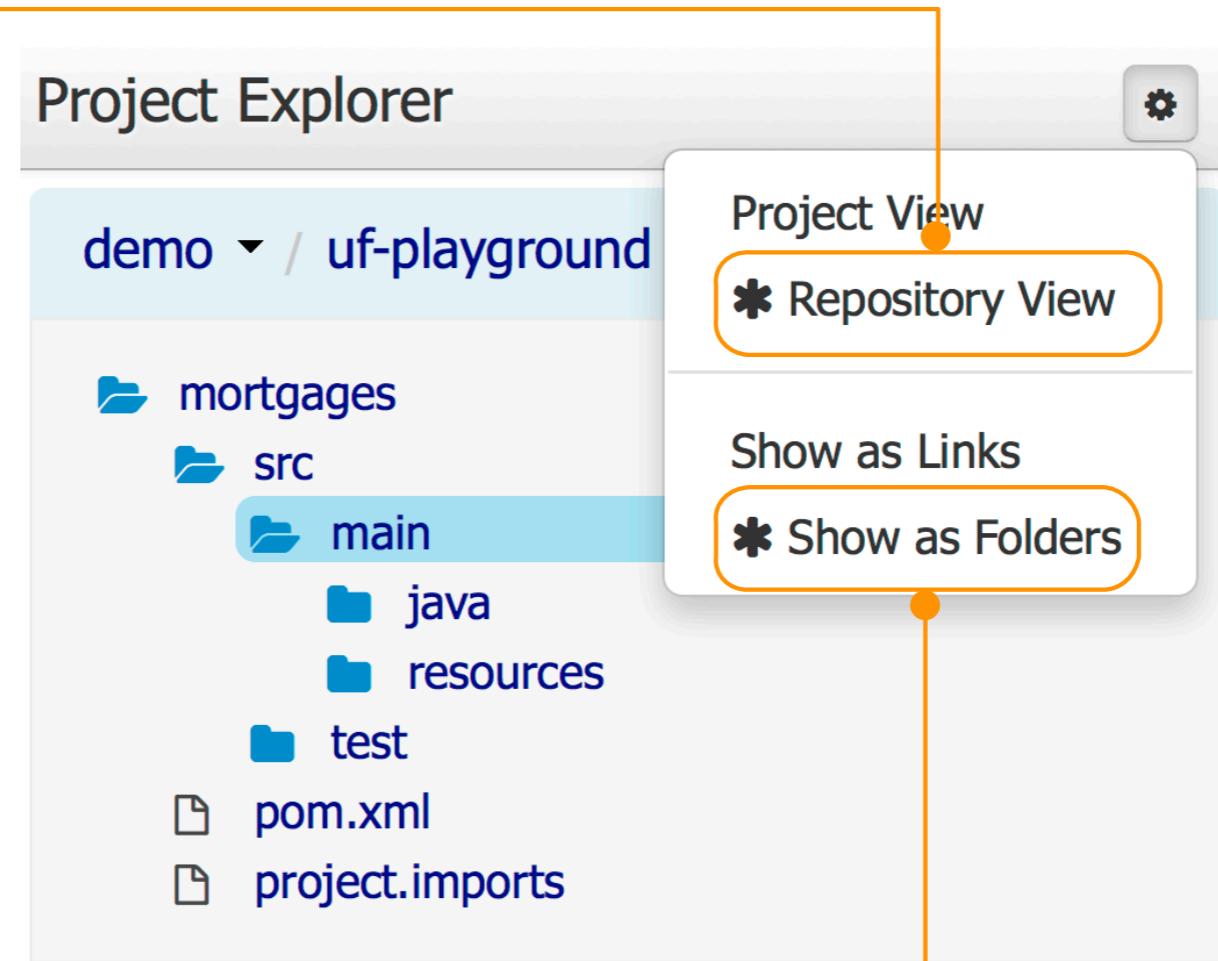
Show as Links

Compact view, that displays the folders as a single line path.

Repository View

Repository View

The repository view provides raw access to the project structure and files, nothing is hidden. And all paths are visible separately; see the java and resources folders.



Show as Folders

Shows the project folders using a tree view.

Show as Links

Compact view, that displays the folders as a single line path.

KIE Workbench

Find User: jbpm ▾

Explore New Item Tools

search...

9

Project Explorer

example ▾ / uf-playground ▾ / mortgages ▾

📁 <default>
📁 org
📁 mo

DRL

© DOMAIN SPECIFIC LANGUAGE DEFINITION

GENUMLATION DEFINITION

GUIDED DECISION TABLE

GUARDED BY E

Bankruptcy history

No bad credit checks

no NINJAs

Underage

GUIDED RULE (WITH DSL)

JAVA SOURCE FILES

Guided Editor [No bad credit checks]

EXTENDS None selected

WHEN

1. There is a LoanApplication [a]
Any of the following are true:
There is an Applicant with
creditRating equal to
2. There is an Applicant with

THEN

1. Retract LoanApplication [app]
Set value of LoanApplication [app]
2. Set value of LoanApplication [app]

Edit Source Config Metadata

Problems

Level **Text** **File** **Column** **Line**

File Edit View History Bookmarks Tools Accessibility Help

Inbox (1) - michael.lanstis@gmail.com KIE Workbench localhost:8080/kie-wb/org.kie.workbench.KIEWebapp/KIEWebapp.html?#default://master@uf-playground/mortgages/src/main/resource: Google

KIE Workbench

Home Authoring Deploy Process Management Tasks Dashboards Find User: jbpm

Explore New Item Tools search...

Project Explorer

- example / uf-playground / mortgages
 - <default>
 - org
 - mortgages

DRL

DOMAIN SPECIFIC LANGUAGE DEFINITION

ENUMERATION DEFINITION

GUIDED DECISION TABLE

GUIDED RULE

- Bankruptcy history
- No bad credit checks
- no NINJAs
- Underage

GUIDED RULE (WITH DSL)

JAVA SOURCE FILES

Guided Editor [Bankruptcy history]

EXTENDS None selected

WHEN

- There is a LoanApplication [a]
The following exists:
There is a Bankruptcy with:
any of the following:
- yearOfOccurrence greater
amountOwed greater

THEN

- Retract LoanApplication [a]
Set value of LoanApplication [a]
- Set value of LoanApplication [a]

(show options...)

Edit Source Config Metadata

Guided Editor [No bad credit checks]

EXTENDS None selected

WHEN

- There is a LoanApplication [app]
Any of the following are true:
There is an Applicant with:
creditRating equal to OK
- There is an Applicant with:
creditRating equal to Sub prime

THEN

- Retract LoanApplication [app]
Set value of LoanApplication [app] approved false
- Set value of LoanApplication [app] explanation Only AA

(show options...)

Edit Source Config Metadata

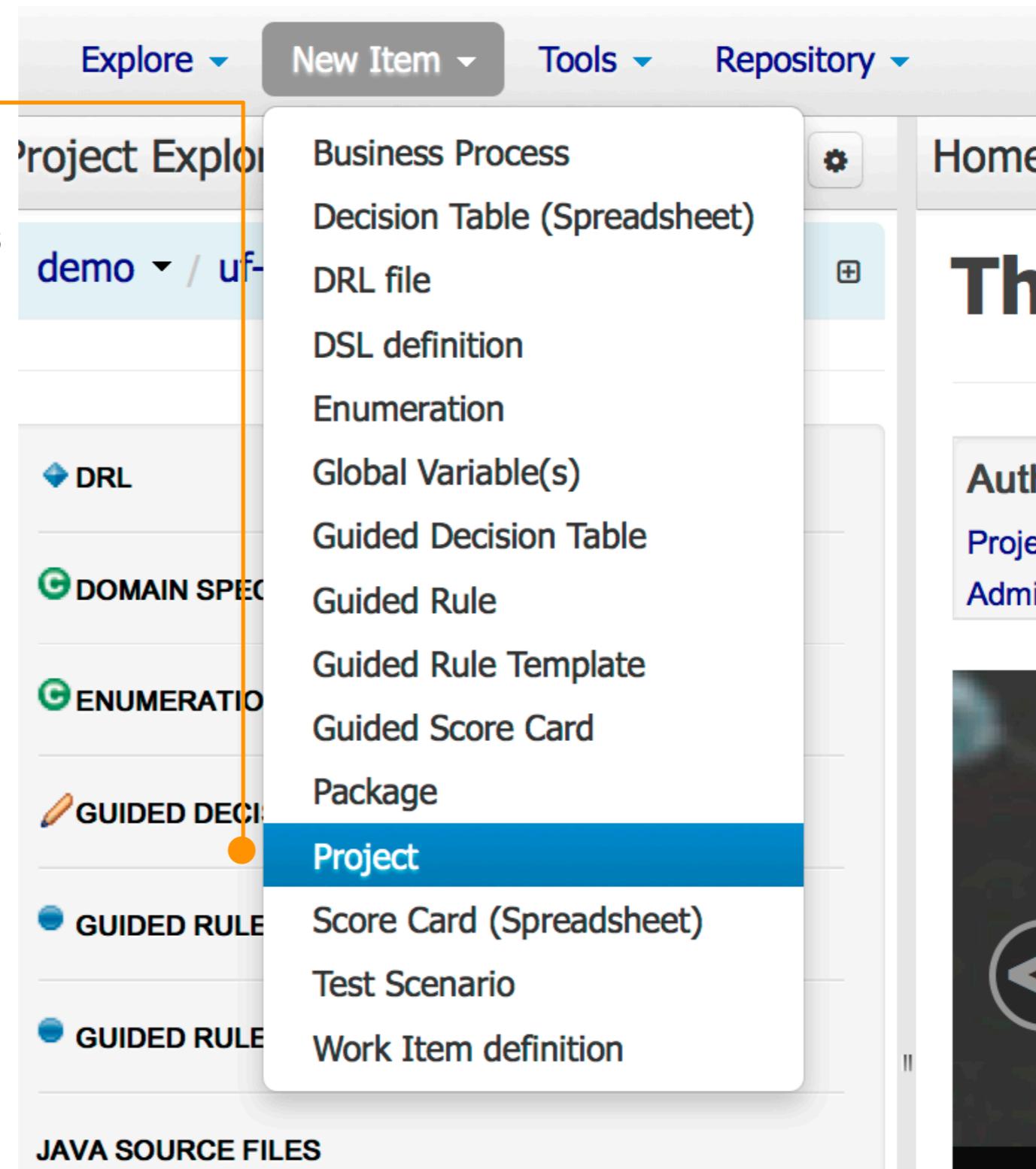
Problems

Level	Text	File	Column	Line
...

New Projects and Other Items

New Projects

Launches the new project wizard. Once created it appears under the project list for the target repository



Project Editor

Project: [human-resources:org.jbpm:1.0]

Save Delete Rename Copy Build & Deploy x ▼

Project General Settings

Project Name

Project Description

Group artifact version

Group ID Example: com.myorganization.myprojects ?

Artifact ID Example: MyProject ?

Version ID 1.0.0 ?

Project Dependencies

Project Screen X

Group ID

Artifact ID

Version ID

Dependencies

Add Add from repository

Group ID	Artifact ID	Version ID	
com.megacorp	hr-model	1.3	
com.megacorp	finance-model	24	

pom.xml pom.xml metadata kmodule.xml kmodule.xml metadata More... ▾

Build and Deploy

Explore ▾ New ▾ Tools ▾ File ▾ **Build & Deploy**

Project Screen ×

Group ID

Artifact ID

Version ID

Dependencies

Group ID	Artifact ID	Version

Repository Manager

Home Perspectives ▾ Logout Explore ▾ Find (Search...)

Guvnor M2 REPOSITORY X

Upload new Jar: [Browse...](#) [upload](#)

Find items with a name matching:

[Search](#)

[Refresh](#) [Delete selected jar](#) [ViewAuditLog](#)

Name	Path	LastModified	View Artifact Detail	Download
<input type="checkbox"/> org.test.mortgages-1.0.jar	org/guvnor/example/org.test.mortgages/1.0/org.test.mortgages-1.0.jar	2013 May 22 11:07:21	Open	Download

1 of 1

Repository Manager

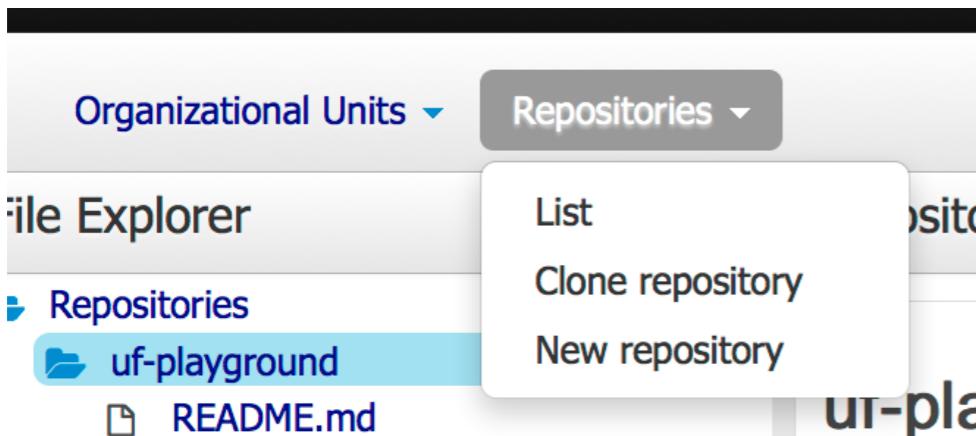
File List				
Name	Path	LastModified	Open	Download
mortgages-0.0.1.jar	mortgages/mortgages/0.0.1 /mortgages-0.0.1.jar	2013 Nov 16 15:46:40	Open	Download
example-1.0.jar	org/drools/example/1.0/example-1.0.jar	2013 Nov 16 15:08:26	Open	Download
jboss-modules-1.1.1.GA.jar	org/jboss/modules/jboss-modules/1.1.1.GA/jboss-modules-1.1.1.GA.jar	2013 Nov 16 15:07:18	Open	Download
async-examples-1.0.jar	org/jbpm/example/async-examples/1.0/async-examples-1.0.jar	2013 Nov 16 16:14:33	Open	Download
HR-1.0.jar	org/jbpm/HR/1.0/HR-1.0.jar	2013 Nov 16 16:14:13	Open	Download

« ‹ › » 1-5 of 5

GIT Repositories

Repositories

Create or clone GIT repositories



Create Repository

A screenshot of a 'Create Repository' form. The title 'Repository Infomation * is required' is displayed at the top. Below it are two fields: 'Repository Name' (marked with a red asterisk) and 'Organizational Unit'. Both fields have placeholder text ('repository name...' and '--- Select ---' respectively) and are enclosed in input boxes.

Clone Repository

A screenshot of a 'Clone Repository' form titled 'Repository Infomation * is required'. It contains three fields: 'Repository Name' (marked with a red asterisk), 'Organizational Unit' (marked with a red asterisk), and 'Git URL' (marked with a red asterisk). Each field has a placeholder text ('repository name...', '--- Select ---', and 'git url...' respectively) and is enclosed in an input box.

Organizational Units

Manage Organizational Units

Create Organizational Units and associate repositories with them.

The screenshot shows the JBoss Repository Manager interface with the 'Organizational Units' tab selected in the top navigation bar. The main area is titled 'Organizational Unit Manager'. On the left, there's a sidebar titled 'Repositories' showing a tree structure with a selected folder 'uf-playground' containing files 'README.md', 'categories.xml', 'mortgages', and 'todo.md'. The main panel has three sections: 'Organizational Units' (containing 'demo : demo@demo.org'), 'Associated repositories' (containing '-- No Organizational Unit selected'), and 'Available repositories' (containing '-- No Organizational Unit selected'). Below these sections are two sets of buttons: 'Add' (blue), 'Edit' (grey), and 'Delete' (red).

End Users - Task Lists

KIE Workbench

Home Authoring Deploy Process Management Tasks Dashboards Find User: salaboy ?

Search... 

Tasks List  Refresh  

Grid Calendar Active Personal Group All

ID	Task	Priority	Status	Created On	Due On	Actions
9	Set up new account with service zxc	0	InProgress	25/02/2014 23:02	26/02/2014 00:00	  
8	Review Meeting Document for Project ABC	0	InProgress	25/02/2014 23:02	26/02/2014 00:00	  
7	Approve Project ABC	0	InProgress	25/02/2014 23:01	26/02/2014 00:00	  
6	Call Customer #123	0	InProgress	25/02/2014 23:01	26/02/2014 00:00	  

Details      

9 - Set up new account with service zxc

Details

Description

Status InProgress

Due On 26/02/2014 23:00

Priority 0 - High

User salaboy

Process Context

End Users - Task Lists

		Grid	Calendar	Active	Personal	Group	All	
Id	Task	Priority	Status	Created On	Due On	Actions		
9	Set up new account with service zxc	0	InProgress	25/02/2014 23:02	26/02/2014 00:00			
8	Review Meeting Document for Project ABC	0	InProgress	25/02/2014 23:02	26/02/2014 00:00			

Details

Description

Status InProgress

Due On 26/02/2014 23:00

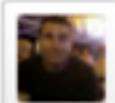
Priority 0 - High

User salaboy

Videos

Mark Proctor

Home Videos Playlists Discussion About 

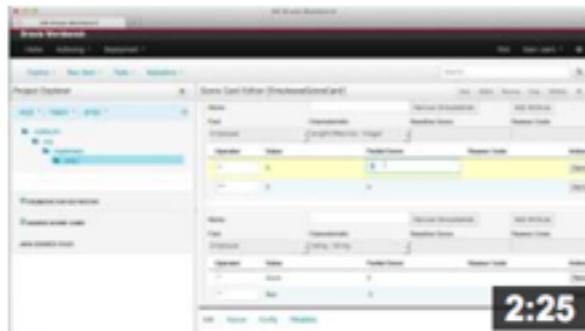


Share your thoughts

All activities ▾

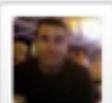


[Mark Proctor](#) uploaded and added to Drools Editor Metaphores

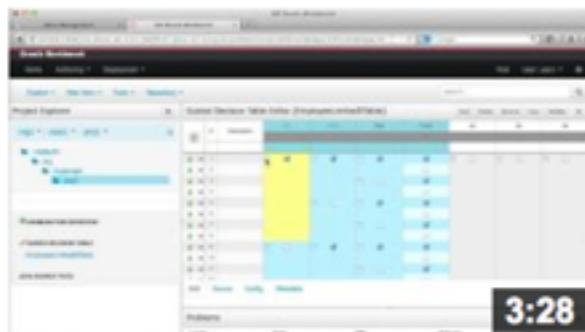


[Score Cards \(Full Screen 720HD\)](#)

2 months ago • 207 views



[Mark Proctor](#) uploaded and added to Drools Editor Metaphores



[Limited Entry Guided Decision \(Full Screen 720HD\)](#)

2 months ago • 255 views

Create a Limited Entry Guided Decision Table

KIE Workbench Videos

http://www.youtube.com/watch?v=vj3MNmiUnvY&index=2&list=PLb9jQNHBKBRj9IJkc_F5nCJAvXaegOGW8

Screenshot of a YouTube playlist titled "KIE Worbench" by Mark Proctor, containing 6 videos.

Index	Thumbnail	Video Title	Uploader	Duration
1		KIE Workbench Installation and Setup (Full Screen 720HD)	Mark Proctor	1:18
2		Authoring Perspective - Project Explorer (Full Screen 720HD)	Mark Proctor	2:11
3		Authoring Perspective - Window System (Full Screen 720HD)	Mark Proctor	1:43
4		Authoring Perspective - Problems and Versioning (Full Screen 720HD)	Mark Proctor	3:15
5		Project Editor and Deployment Perspective (Full Screen 720HD)	Mark Proctor	1:52
6		Organization Unit and Repository Administration Perspective (Full Screen 720HD)	Mark Proctor	4:48

Drools Editor Videos

<http://www.youtube.com/playlist?list=PLb9jQNHBKRipbtadRC-UaUObjwp0aBHJ>

 **Drools Editor Metaphores**
by Mark Proctor • 7 videos • 979 views • 15 minutes

Add a description Play all Share Playlist settings Add video

#	Video Title	Uploader	Duration
1	Create a Project (Full Screen 720HD)	Mark Proctor	1:09
2	Create a Model (Full Screen 720HD)	Mark Proctor	1:21
3	Limited Entry Guided Decision (Full Screen 720HD)	Mark Proctor	3:28
4	Extended Entry Guided Decision Table (Full Screen 720HD)	Mark Proctor	2:24
5	Rule Template (Full Screen 720HD)	Mark Proctor	1:53
6	Rule Fragment in Guided Decision Table (Full Screen 720HD)	Mark Proctor	3:16
7	Score Cards (Full Screen 720HD)	Mark Proctor	2:25

KIE Drools Workbench

JBoss Management KIE Drools Workbench +

Drools Workbench

Home Authoring Deployment Find User: user1 ?

Explore New Item Tools Repository search... Q

Project Explorer

demo / uf-playground / mortgages

<default>

org

mortgages

DRL

Dummy rule

DOMAIN SPECIFIC LANGUAGE DEFINITION

ENUMERATION DEFINITION

GUIDED DECISION TABLE

Pricing loans

GUIDED RULE

GUIDED RULE (WITH DSL)

DRL Editor [Dummy rule]

Show fact types

```
package org.mortgages

rule 'Dummy rule'

when
    //conditions
then
    //actions

end
```

Save Delete Rename Copy Validate X

DRL Metadata

Recently Edited

Refresh Refresh

Format	Name	Created Date	Open
document	Pricing loans.gdst	2013 Nov 20 20:32:46	Open



Project Explorer



uf-playground.git /



GuvnorM2RepoDependencyExample1

GuvnorM2RepoDependencyExample2

mortgages

tonissandbox

bpmn2

defaultPackage

Recently Edited



Refresh Open selected Delete selected



Created

Format Name Date Open

Dummy rule.drl 2013 May 22
11:07:12 Open

Problems



Level

Text

File

Artifact ID

Column

Line

Build Deploy Utilize

API

BRMS 5.0 Programmatic API

```
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();

kbuilder.batch().add( newClassPathResource( "Model.drl", getClass() ), DRL )
    .add( newClassPathResource( "Queries.drl", getClass() ), DRL )
    .add( newClassPathResource( "General.drl", getClass() ), DRL )
    .add( newClassPathResource( "Response.drl", getClass() ), DRL )
    .add( newClassPathResource( "Events.drl", getClass() ), DRL )
    .add( newClassPathResource( "UiView.drl", getClass() ), DRL )
    .add( newClassPathResource( "Commands.drl", getClass() ), DRL ).build()

if ( kbuilder.hasErrors() ) {
    System.out.println( kbuilder.getErrors().toString() );
    System.exit( 1 );
}

KieBaseConfiguration kbaseConf = KnowledgeBaseFactory.newKnowledgeBaseConfiguration();
kbaseConf.setOption( EqualityBehaviorOption.EQUALITY );

KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase( kbaseConf );
kbase.addKnowledgePackages( kbuilder.getKnowledgePackages() );

Counter c = new Counter();
ksession = kbase.newStatefulKnowledgeSession();
```

KieModules

- Discovery
 - META-INF/kmodule.xml
- Convention based
- No programmatic api for building
- Multiple Named entities
- Inheritance of Resources
- Defaults for lazy people
- Version built in a standard

GitHub Examples

- <https://github.com/droolsjbpm/drools/tree/master/drools-examples>

drools/drools-examples-api at master · droolsjbpm/drools · GitHub

GitHub, Inc. GitHub.com/droolsjbpm/drools/tree/master/drools-examples-api Reader Import to Mendeley

Google UK Froogle UK 100 Greatest...tar Albums Answering M...m the Past) Create a stu...igner Depot NEXT – Nati... and HTML5 Import to Mendeley > +

to version 6.0.0-SNAPSHOT

mbiarnes authored a month ago latest c

..

File	Last Commit	Description
default-kiesession-from-file	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
default-kiesession	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
kie-module-from-multiple-files	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
kiebase-inclusion	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
kiecontainer-from-kierepo	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
kiefilesystem-example	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
kiemodulemodel-example	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
named-kiesession-from-file	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
named-kiesession	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]
.gitignore	4 months ago	add kie api examples [mariofusco]
pom.xml	a month ago	to version 6.0.0-SNAPSHOT [mbiarnes]

KieModules

```
<kmodule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://jboss.org/kie/6.0.0/kmodule">

</kmodule>
```

```
KieServices ks = KieServices.Factory.get();
KieContainer kContainer = ks.getKieClasspathContainer();

KieSession kSession = kContainer.newKieSession();
kSession.setGlobal("out", out);
kSession.insert(new Message("Dave", "Hello, HAL. Do you read me, HAL?"));
kSession.fireAllRules();
```

KieModules

- Named Entities and JAR on Classpath
- Creates one KieBase
- Includes resources from package matching kbase name

```
<kmodule xmlns="http://jboss.org/kie/6.0.0/kmodule">  
  
    <kbase name="kbase1">  
        <ksession name="ksession1"/>  
    </kbase>  
  
</kmodule>
```

```
KieServices ks = KieServices.Factory.get();  
KieContainer kContainer = ks.getKieClasspathContainer();  
  
KieSession kSession = kContainer.newKieSession("ksession1");  
kSession.setGlobal("out", out);  
kSession.insert(new Message("Dave", "Hello, HAL. Do you read me, HAL?"));  
kSession.fireAllRules();
```

KieModules

- Named Entities, with inheritance and JAR on Classpath
- Two projects, one “includes” from the other

```
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>named-kiesession</artifactId>
  <version>6.0.0-SNAPSHOT</version>
</dependency>
```

```
<kbase name="kbase2" includes="kbase1">
  <ksession name="ksession2"/>
</kbase>
```

```
KieServices ks = KieServices.Factory.get();
KieContainer kContainer = ks.getKieClasspathContainer();
KieSession kSession = kContainer.newKieSession("ksession2");
kSession.setGlobal("out", out);

kSession.insert(new Message("Dave", "Hello, HAL. Do you read me, HAL?"));
kSession.fireAllRules();

kSession.insert(new Message("Dave", "Open the pod bay doors, HAL."));
kSession.fireAllRules();
```

KieModules

- Package location can over-ride kbase name defaults

```
<kbase name="WumpusMainKB" packages="org.drools.games.wumpus.server,  
                                org.drools.games.wumpus.server.view">  
    <ksession name="WumpusMainKS" />  
</kbase>  
  
<kbase name="WumpusClientKB" packages="org.drools.games.wumpus.client">  
    <ksession name="WumpusClientKS"/>  
</kbase>
```

```
KieContainer kc = KieServices.Factory.get().getKieClasspathContainer();  
final KieSession serverKsession = kc.newKieSession( "WumpusMainKS" );  
final KieSession clientKsession = kc.newKieSession( "WumpusClientKS" );
```

Dynamic KieModules

- JARs can be loaded from URLs into KieRepository
- Once loaded they can be resolved via ReleaseId

```
KieServices ks = KieServices.Factory.get();
KieRepository kr = ks.getRepository();

KieModule kModule = kr.addKieModule(ks.getResources().newFileSystemResource(
    getFile("default-kiesession")));
    
KieContainer kContainer = ks.newKieContainer(kModule.getReleaseId());

KieSession kSession = kContainer.newKieSession();
kSession.setGlobal("out", out);

Object msg1 = createMessage(kContainer,
                            "Dave", "Hello, HAL. Do you read me, HAL?");
kSession.insert(msg1);
kSession.fireAllRules();
```

Dynamic KieModules

- kie-ci use embedded maven for remote discovery

```
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>
```

```
KieServices ks = KieServices.Factory.get();

// Install example1 in the local maven repo before to do this
KieContainer kContainer = ks.newKieContainer(
    ks.newReleaseId("org.drools",
                    "named-kiesession",
                    "6.0.0-SNAPSHOT"));

KieSession kSession = kContainer.newKieSession("ksession1");
kSession.setGlobal("out", out);

Object msg1 = createMessage(kContainer,
                           "Dave", "Hello, HAL. Do you read me, HAL?");
kSession.insert(msg1);
kSession.fireAllRules();
```

Dynamic KieModules

```
// create a new kjar
InternalKieModule kJar2 = createKieJar(ks, releaseId, "rule2", "rule3");

// deploy it on maven
repository.deployArtifact(releaseId, kJar2, kPom);

// since I am not calling start() on the scanner it means it won't have
// automatic scheduled scanning
KieScanner scanner = ks.newKieScanner(kieContainer);

// scan the maven repo to get the new kjar version and deploy it on the
// kcontainer
scanner.scanNow();

// create a ksesion and check it works as expected
KieSession ksession2 = kieContainer.newKieSession("KSession1");
checkKSession(ksession2, "rule2", "rule3");
```

Maven Versions

- <version>1.0.1</version>
- <version>[1.0.0,2.0.0)</version>
- <version>[1.0.0,)</version>
- <version>LATEST</version>
- <version>RELEASE</version>
- <version>SNAPSHOT</version>

kmodule.xml Editor

Project Screen X

KBases

Add Rename Delete Make Default

Name

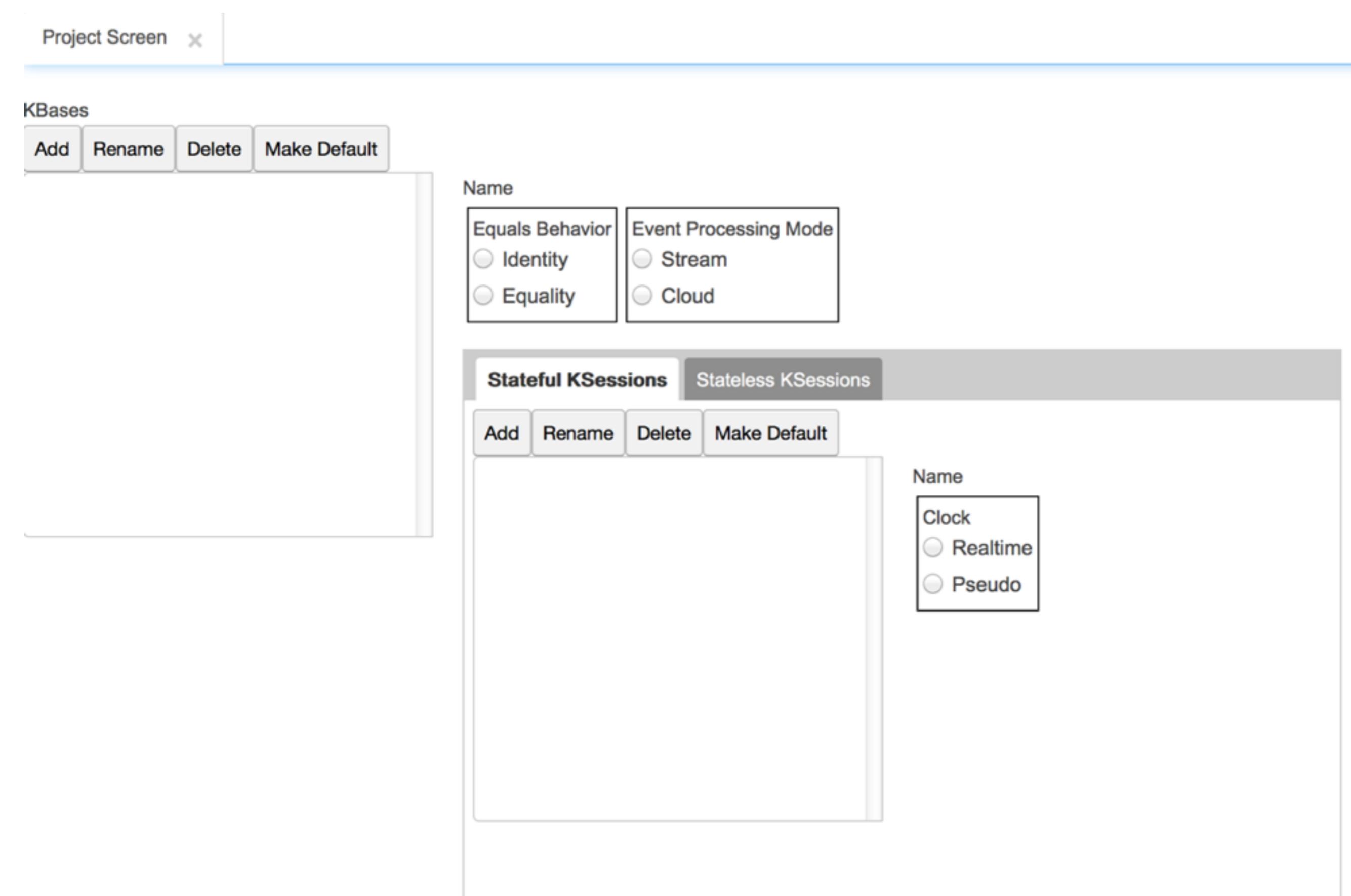
Equals Behavior	Event Processing Mode
<input type="radio"/> Identity	<input type="radio"/> Stream
<input type="radio"/> Equality	<input type="radio"/> Cloud

Stateful KSessions Stateless KSessions

Add Rename Delete Make Default

Name

Clock
<input type="radio"/> Realtime
<input type="radio"/> Pseudo



The screenshot displays the 'kmodule.xml Editor' interface. At the top left is the 'Project Screen' tab, which is currently active. Below it is the 'KBases' section, containing buttons for 'Add', 'Rename', 'Delete', and 'Make Default'. Under 'Name', there are two boxes: one for 'Equals Behavior' with options 'Identity' and 'Equality', and another for 'Event Processing Mode' with options 'Stream' and 'Cloud'. The 'Event Processing Mode' box has 'Cloud' selected. Below the KBases section is a large panel titled 'Stateful KSessions' and 'Stateless KSessions', with 'Stateless KSessions' being the active tab. This panel also contains 'Add', 'Rename', 'Delete', and 'Make Default' buttons. It features a 'Name' section with a single box for 'Clock', which has 'Realtime' selected. The 'Clock' box has 'Pseudo' as an option.

Programmatic API

- Builder API, for tooling integration
- Incremental compilation, and problem reporting

```
KieServices ks = KieServices.Factory.get();
KieRepository kr = ksgetRepository();
KieFileSystem kfs = ks.newKieFileSystem();

kfs.write("src/main/resources/org/kie/example5/HAL5.drl", getRule());

KieBuilder kb = ks.newKieBuilder(kfs);

kb.buildAll(); // kieModule is automatically deployed to KieRepository if
successfully built.
if (kb.getResults().hasMessages(Level.ERROR)) {
    throw new RuntimeException("Build Errors:\n" + kb.getResults().toString());
}

KieContainer kContainer = ks.newKieContainer(kr.getDefaultReleaseId());

KieSession kSession = kContainer.newKieSession();
kSession.setGlobal("out", out);

kSession.insert(new Message("Dave", "Hello, HAL. Do you read me, HAL?"));
kSession.fireAllRules();
```

CDI

CDI Context and Dependency

- CDI injects named entities from the kmodule.xml
- Injectable types
- KieServices
- KieContainer
- KieBase
- KieSession
- StatelessKieSession

KBase

```
@Inject  
private KieBase defaultClassPathKBase;
```

```
@Inject  
@KReleaseId( groupId = "jar1",  
               artifactId = "art1",  
               version = "1.0")  
private KieBase defaultDynamicKBase;
```

```
@Inject  
@KBase(value="jar1.KBase1", name="kb2")  
@KReleaseId( groupId = "jar1",  
               artifactId = "art1",  
               version = "1.0")  
private KieBase jar1KBase1kb2;
```

```
@Inject  
@KBase(value="jar1.KBase1", name="kb2")  
@KReleaseId( groupId = "jar1",  
               artifactId = "art1",  
               version = "1.0")  
private KieBase jar1KBase1kb22;
```

```
@Inject  
@KBase("jar1.KBase1")  
@KReleaseId( groupId = "jar1",  
               artifactId = "art1",  
               version = "1.0")  
private KieBase jar1KBase1v10;
```

```
@Inject  
@KBase("jar1.KBase1")  
@KReleaseId(groupId = "jar1",  
               artifactId = "art1",  
               version = "1.1")  
private KieBase jar1KBase1v11;
```

```
@Inject  
@KBase(value="jar1.KBase1", name="kb1")  
@KReleaseId( groupId = "jar1",  
               artifactId = "art1",  
               version = "1.0")  
private KieBase jar1KBase1kb1;
```

KSession

```
@Inject  
@KSession("jar1.KSession2")  
@KReleaseId( groupId = "jar1",  
              artifactId = "art1",  
              version = "1.0" )  
private KieSession kbase1ksession2v10;  
  
@Inject  
@KSession("jar1.KSession2")  
@KReleaseId( groupId = "jar1",  
              artifactId = "art1",  
              version = "1.1" )  
private KieSession kbase1ksession2v11;
```

```
@Inject  
@KSession(value="jar1.KSession2", name="ks1")  
@KReleaseId( groupId = "jar1",  
              artifactId = "art1",  
              version = "1.0" )  
private KieSession kbase1ksession2ks1;  
  
@Inject  
@KSession(value="jar1.KSession2", name="ks2")  
@KReleaseId( groupId = "jar1",  
              artifactId = "art1",  
              version = "1.0" )  
private KieSession kbase1ksession2ks2 ;  
  
@Inject  
@KSession(value="jar1.KSession2", name="ks2")  
@KReleaseId( groupId = "jar1",  
              artifactId = "art1",  
              version = "1.0" )  
private KieSession kbase1ksession2ks22;
```

Spring and Camel

Spring and Camel

```
<kie:kmodule id="CxfRsSpring">
  <kie:kbase name="test1" packages="test1">
    <kie:ksession name="ksession1">
      <kie:batch>
        <kie:set-global identifier="list" >
          <bean class="java.util.ArrayList" />
        </kie:set-global>
      </kie:batch>
    </kie:ksession>
    <kie:ksession name="ksession2"/>
  </kie:kbase>
</kie:kmodule>
```

Spring and Camel

```
<bean id="kiePolicy" class="org.kie.camel.component.KiePolicy" />

<camelContext id="camel" xmlns="http://camel.apache.org/schema/spring">

    <route>
        <from uri="cxfrs://bean://rsServer"/>
        <policy ref="kiePolicy">
            <unmarshal ref="xstream" />
            <to uri="kie:ksession1" />
            <marshal ref="xstream" />
        </policy>
    </route>

    <route id="x1">
        <from uri="direct://http"/>
        <policy ref="kiePolicy">
            <to uri="cxfrs://http://localhost:58001/rest"/>
        </policy>
    </route>

</camelContext>
```

Questions?



- Dave Bowman: All right, HAL; I'll go in through the emergency airlock.
- HAL: Without your space helmet, Dave, you're going to find that rather difficult.
- Dave Bowman: HAL, I won't argue with you anymore! Open the doors!
- HAL: Dave, this conversation can serve no purpose anymore. Goodbye.

Joshya: Greetings, Professor Falken.

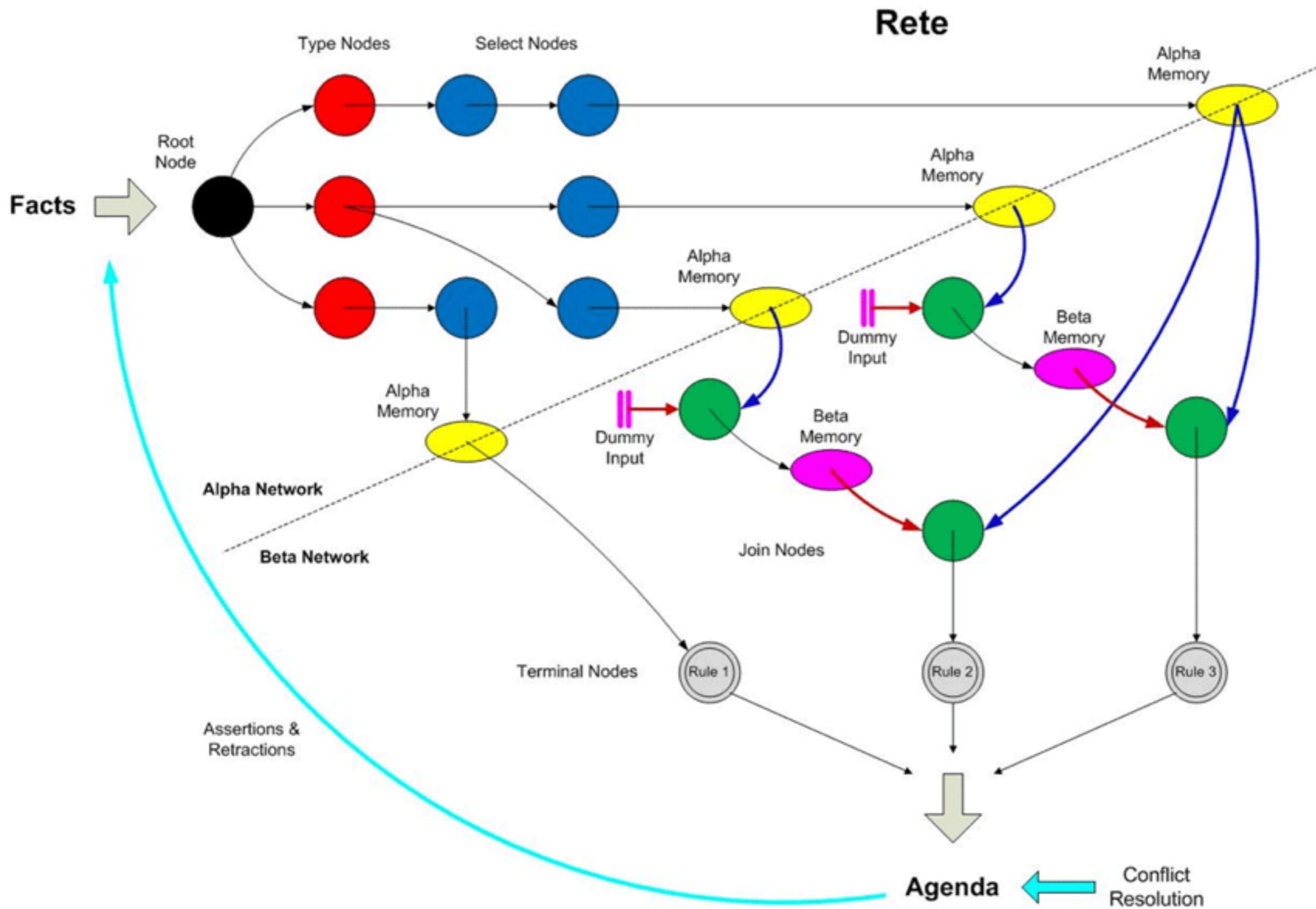
Falken: Hello, Joshua.

Joshya: A strange game. The only winning move is not to play. How about a nice game of chess?

R.I.P Rete
Time to get Phreaky

Rete

Rete – in-memory network



ReteOO

- Node sharing
- Alpha indexing
- Tree based graphs
- Modify-in-place
- Property reactive
- Sub-networks
- Backward Chaining
- Lazy Truth Maintenance
- Heap based agenda
- Dynamic Rules

R.I.P RETE Time to get Phreaky

inspirations:

- Leaps, Collection Oriented Match, L/R Unlinking

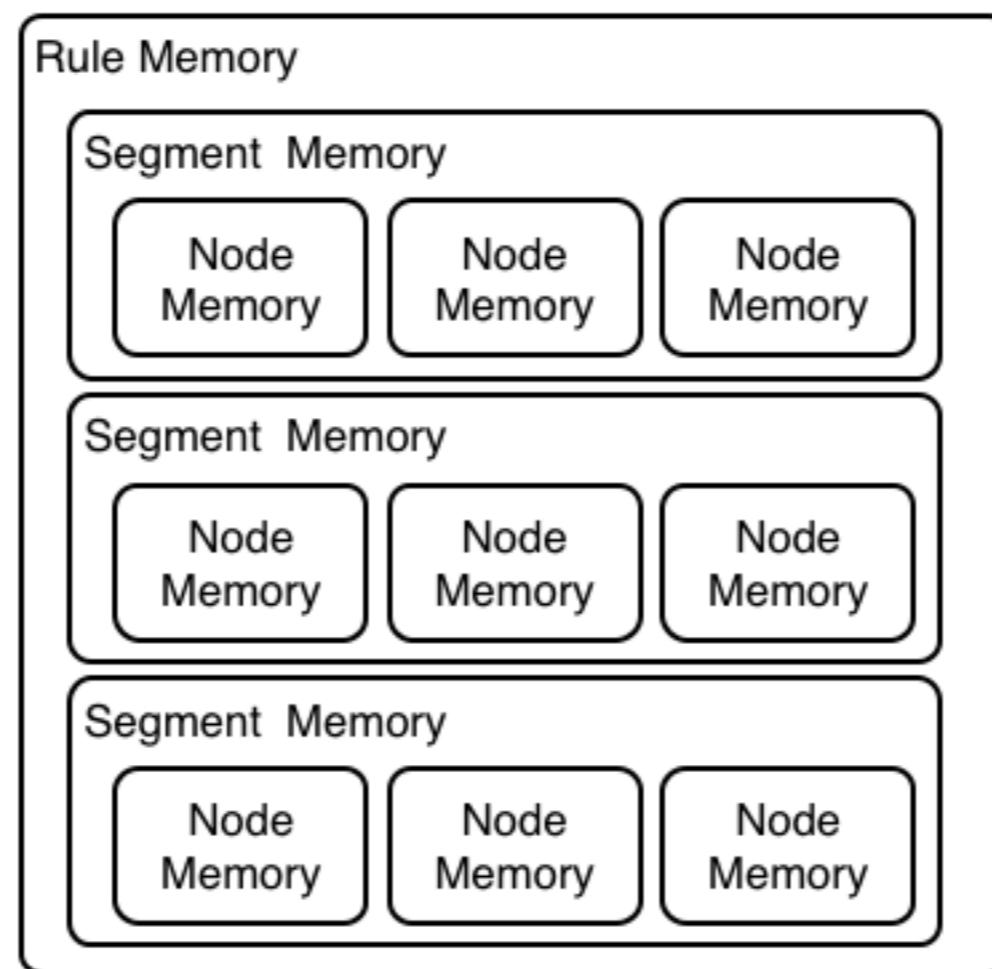
New Innovations

- Full Rule, and Rule Segment Unlinking
- Lazy Evaluation, with Rule scoping
- Set propagations

Previous Innovations

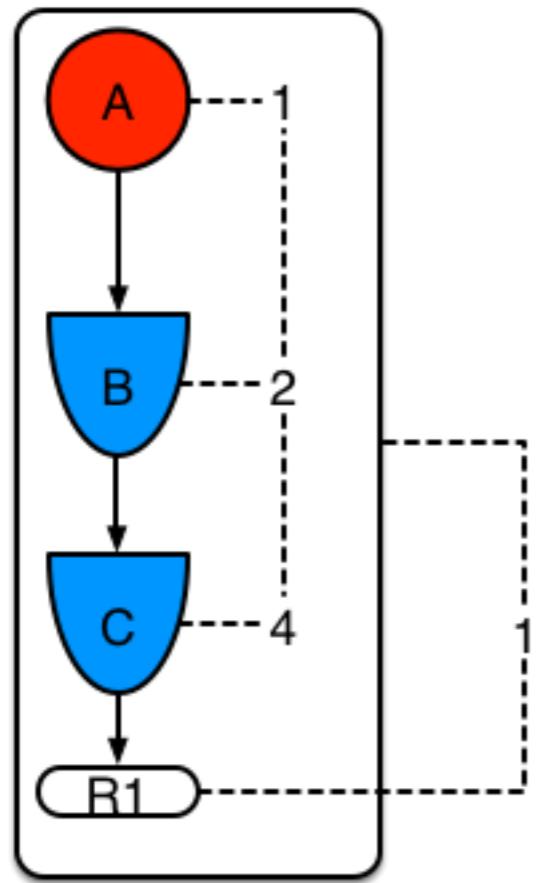
- Modify In Place
- Property Reactive
- Tree Based Graphs
- Subnetwork support

Phreak



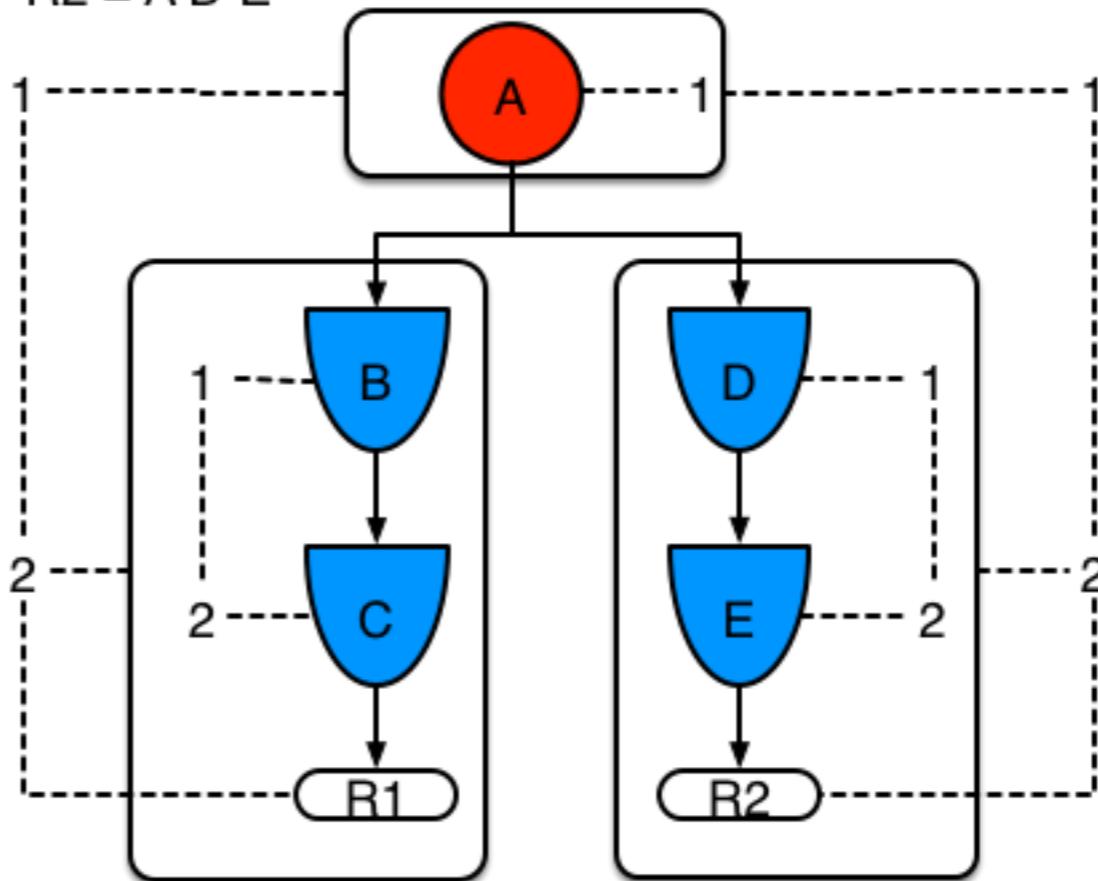
Phreak

R1 = A B C



R1 = A B C

R2 = A D E



Rule Memory

Segment Memory

Node
Memory

Node
Memory

Node
Memory

Segment Memory

Node
Memory

Node
Memory

Node
Memory

Segment Memory

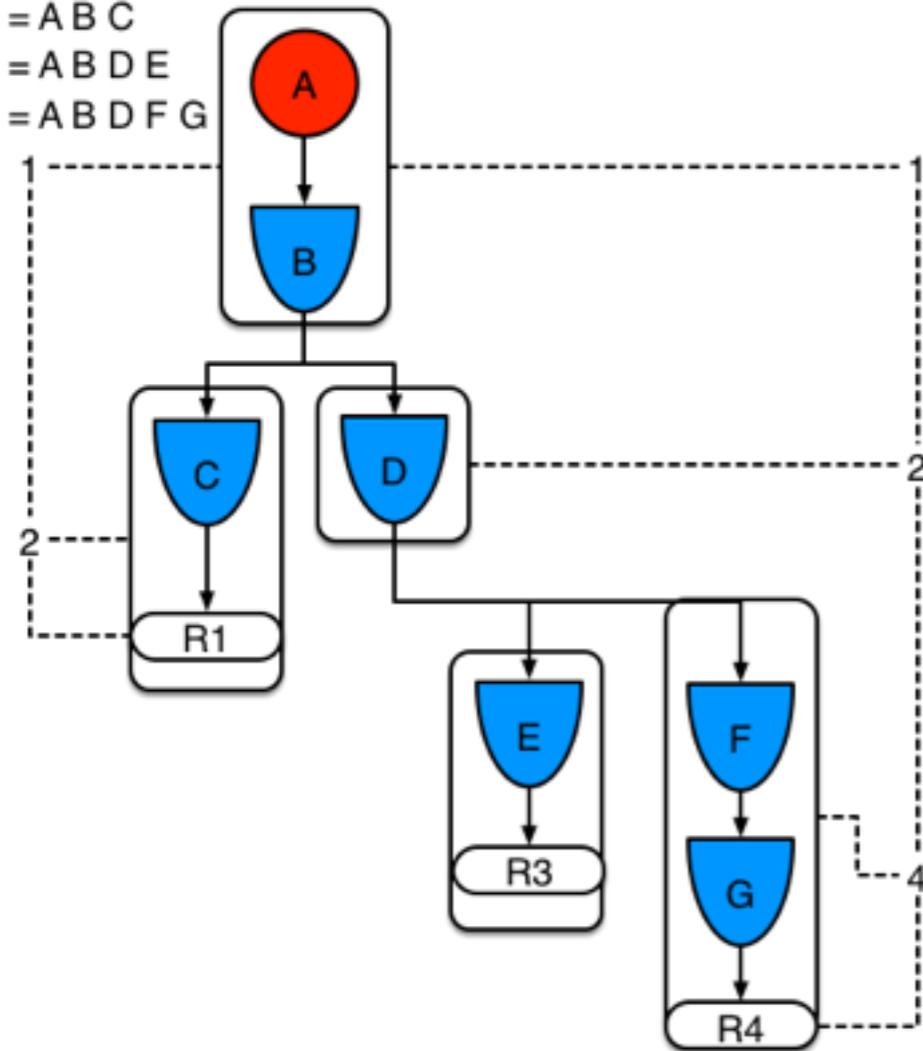
Node
Memory

Node
Memory

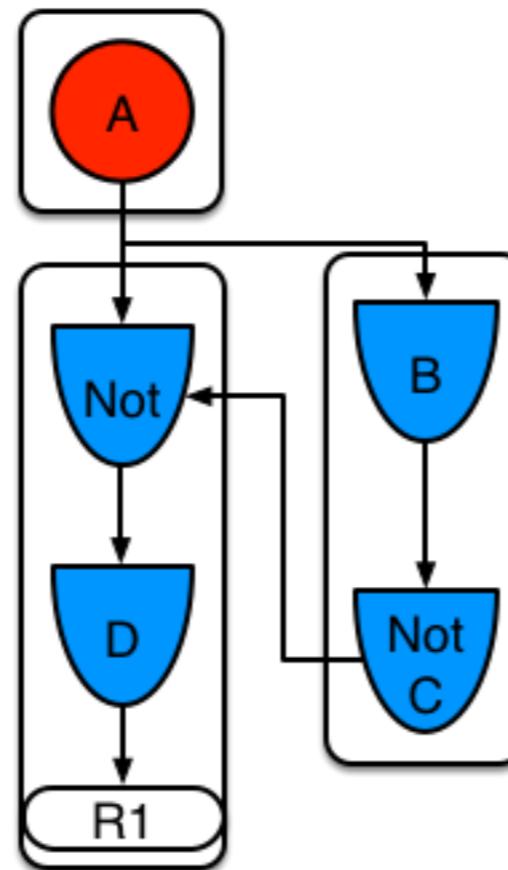
Node
Memory

Phreak

R1 = A B C
R3 = A B D E
R4 = A B D F G

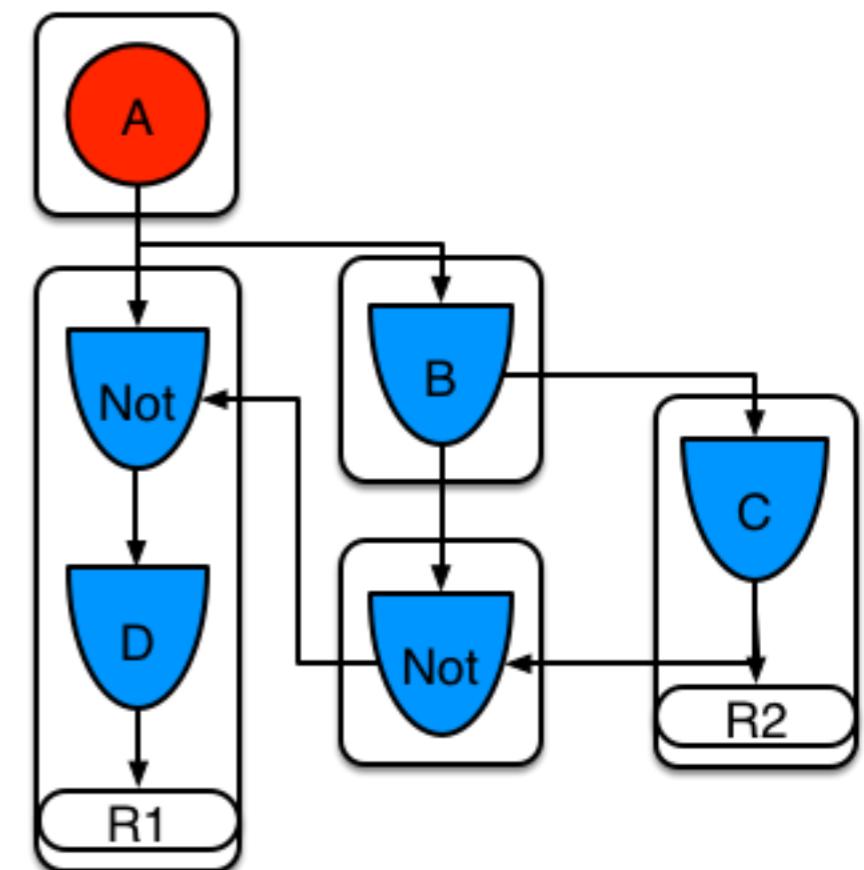


R1 = A not (B not (C)) D



R1 = A not (B not (C)) D

R2 = A B C



TMS

Justification-based Truth

- Drools 5.x
- Simple logical insertion TMS, like Clips, Jess and others.
- Drools 6.0
- Contradiction handling with JTMS
 - Clean separation of exception logic
- TMS now has pluggable Belief System
 - Simple TMS support
 - JTMS now possible (exp)
 - Defeasible Logic (exp)
- See drools-compiler
 - JTMSTest for lots of example tests

TMS

```
rule "Issue Child Bus Pass"  
when  
    $p : Person( age < 16 )  
then  
    insert(new ChildBusPass( $p ) );  
end  
  
rule "Issue Adult Bus Pass"  
when  
    $p : Person( age >= 16 )  
then  
    insert(new AdultBusPass( $p ) );  
end
```

Couples the logic

What happens when the Child stops being 16?

TMS

- Bad
- Monolithic
- Leaky
- Brittle integrity - manual maintenance

TMS

- A rule “logically” inserts an object
- When the rule is no longer true, the object is retracted.

rule "IsChild"

when

\$p : Person(age < 16)

then

logicalInsert(new IsChild(\$p))

end

rule "IsAdult"

when

\$p : Person(age >= 16)

then

logicalInsert(new IsAdult(\$p))

end

de-couples the logic

Maintains the truth by
automatically retracting

TMS

```
rule "Issue Child Bus Pass"
```

```
when
```

```
    $p : Person()
```

```
        IsChild( person =$p )
```

```
then
```

```
    logicalInsert(new ChildBusPass( $p ) );
```

```
end
```

```
rule "Issue Adult Bus Pass"
```

```
when
```

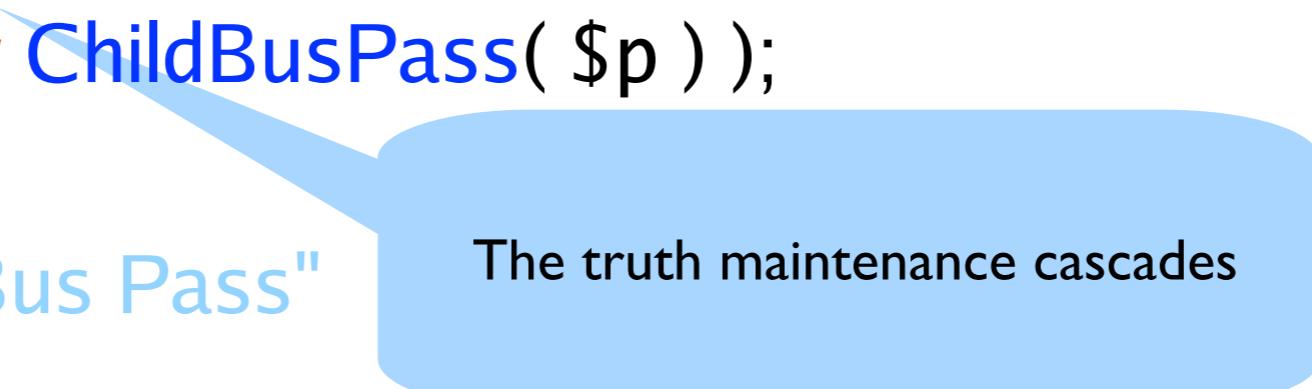
```
    $p : Person()
```

```
        IsAdult( person =$p )
```

```
then
```

```
    logicalInsert(new AdultBusPass( $p ) );
```

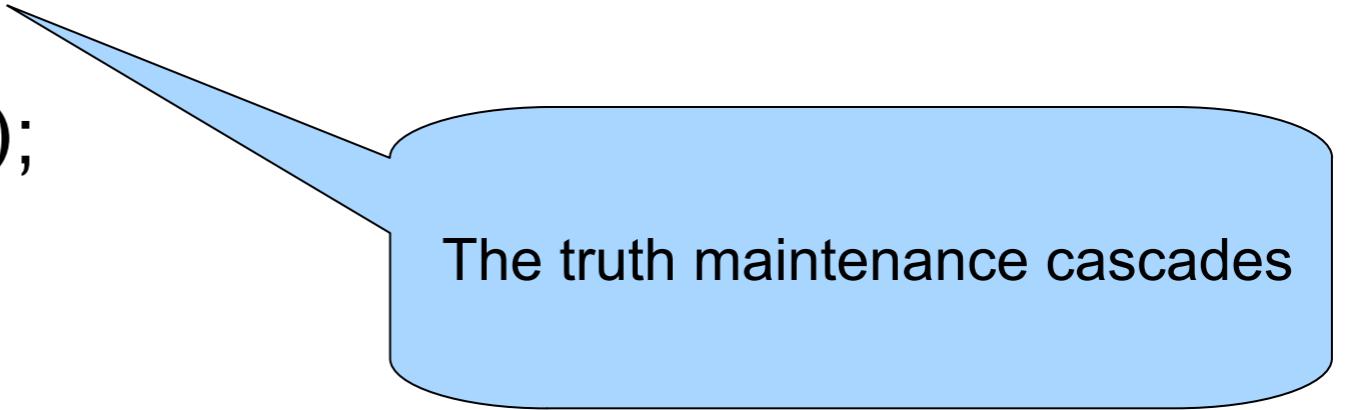
```
end
```



The truth maintenance cascades

TMS

```
rule "Issue Child Bus Pass"  
when  
    $p : Person()  
        not( ChildBusPass( person == $p ) )  
then  
    requestChildBusPass( $p );  
end
```



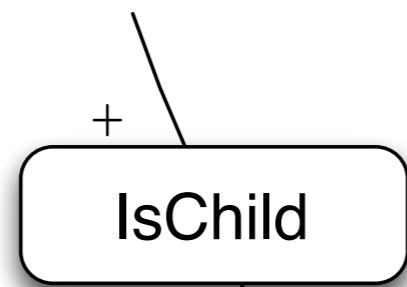
The truth maintenance cascades

TMS

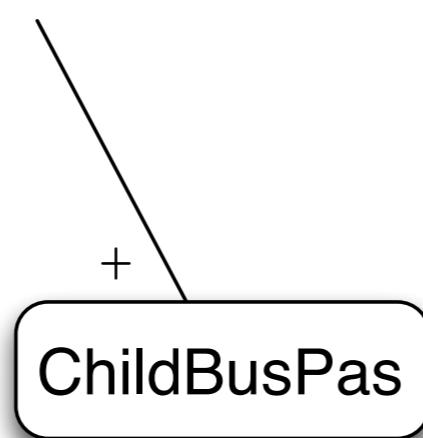
- Good
- De-couple knowledge responsibilities
- Encapsulate knowledge
- Provide semantic abstractions for those encapsulation
- Integrity robustness – truth maintenance

TMS

Rule : isChildRule



Rule : IssueBusPas



JTMS

JTMS

rule "Do not issue to banned people"

when

\$p : Person()

Banned(person =\$p)

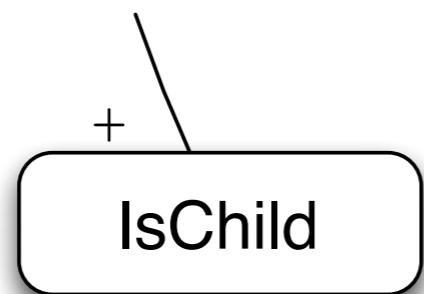
"plus" and "neg"
labels

then

logicalInsert(new ChildBusPass(\$p), "neg");

end

Rule : isChildRule



Rule : Do Not Issue
to Banned People

Rule : IssueBusPas



Defeasible

Defeasible

```
rule "All Birds Fly"  
when  
  $b : Bird()  
then  
  logicalInsert(new Fly( $b ));  
end  
rule "Brocken wing"  
when  
  $b : Bird()  
  BrockenWing($b;)  
then  
  logicalInsert(new Fly( $b ),  
    "neg");  
end
```

Defeasible

```
rule "All Birds Fly" @Defeasible
when
  $b : Bird()
then
  logicalInsert(new Fly( $b ) );
end
rule "Brocken wing" @Defeasible
@Defeater
when
  $b : Bird()
    BrockenWing($b;)
then
  logicalInsert(new Fly( $b ), "neg");
end
```

```
rule "Birds With Rockets Fly" @Defeasible
@Defeats("Brocken Wing")
when
  $b : Bird()
    Rocket($b;)
then
  logicalInsert(new Fly( $b ) );
```

Decision Tables

Decision Table

RuleTable Pricing bracket					
	CONDITION	CONDITION	CONDITION	CONDITION	ACTION
	Driver			policy: Policy	
	age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type	policy.setBasePrice(\$param); System.out.println("Sparam");
Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD
Young safe package	18, 24	LOW	1	COMPREHENSIVE	450
		MED		FIRE_THEFT	200
		MED	0	COMPREHENSIVE	300
		LOW		FIRE_THEFT	150
		LOW	0	COMPREHENSIVE	150
					Safe driver discount
Young risk	18,24	MED	1	COMPREHENSIVE	700
	18,24	HIGH	0	COMPREHENSIVE	700
	18,24	HIGH		FIRE THEFT	550
					Location risk
Mature drivers	25,30		0	COMPREHENSIVE	120
	25,30		1	COMPREHENSIVE	300
	25,30		2	COMPREHENSIVE	590
	25,35		3	THIRD PARTY	800
					High risk

Decision Table

RuleTable Pricing bracket					
CONDITION	CONDITION	CONDITION	CONDITION	ACTION	ACTION
Driver			policy: Policy		
age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type	policy.setBasePrice(\$param);	System.out.println("Param");
Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD
					Record Reason

RuleTable Pricing bracket					
CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	CONDITION
Driver				policy: Policy	
age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type		
Age Bracket	Location risk profile	Number of prior claims	Policy type applying for		

Young risk		LOW	0	COMPREHENSIVE	150	Safe driver discount
	18,24	MED	1	COMPREHENSIVE	700	
	18,24	HIGH	0	COMPREHENSIVE	700	Location risk
	18,24	HIGH		FIRE THEFT	550	Location risk
Mature drivers	25,30		0	COMPREHENSIVE	120	Cheapest possible
	25,30		1	COMPREHENSIVE	300	
	25,30		2	COMPREHENSIVE	590	
	25,35		3	THIRD PARTY	800	High risk

Decision Table

RuleTable Pricing bracket					
CONDITION	CONDITION	CONDITION	CONDITION	ACTION	ACTION
Driver			policy: Policy		
age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type	policy.setBasePrice(\$param);	System.out.println("\$param");
Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD
					Record Reason

RuleTable Pricing bracket					
CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	CONDITION
Driver				policy: Policy	
age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type		
Age Bracket	Location risk profile	Number of prior claims	Policy type applying for		

Young risk	18..24	ACTION	ACTION
	18..24		
	18..24	policy.setBasePrice(\$param);	System.out.println("\$param");
Mature drivers	25..30	Base \$ AUD	Record Reason
	25..30	1	COMPREHENSIVE
	25..30	2	COMPREHENSIVE
	25..35	3	THIRD PARTY
			High risk

Decision Table

RuleTable Pricing bracket						
	CONDITION	CONDITION	CONDITION	CONDITION	ACTION	ACTION
	Driver			policy: Policy		
	age >= \$1, age <= \$2	locationRiskProfile	priorClaims	type	policy.setBasePrice(\$param);	System.out.println("\$param");
Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD	Record Reason
Young safe package	18, 24	LOW	1	COMPREHENSIVE	450	
		MED		FIRE_THEFT	200	Priors not relevant
		MED	0	COMPREHENSIVE	300	

rule "Pricing bracket_10"

when

```
Driver(age >= 18, age <= 24,  
      locationRiskProfile == "LOW",  
      priorClaims == "1")
```

policy:

```
Policy(type == "COMPREHENSIVE")
```

then

```
  policy.setBasePrice(450);
```

end

Decision Table

	CONDITION Driver	CONDITION locationRiskProfile	CONDITION priorClaims	CONDITION policy: Policy
Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for
Young safe package	18, 24	LOW	1	COMPREHENSIVE
		MED		FIRE_THEFT
when		MED	0	COMPREHENSIVE

```
Driver(age >= 18, age <= 24,
       locationRiskProfile == "LOW",
       priorClaims == "1")
```

policy:

```
Policy(type == "COMPREHENSIVE")
```

then

```
    policy.setBasePrice(450);
```

end

Decision Table

	CONDITION	CONDITION	ACTION	ACTION
	Driver age >= \$1, age <= \$2	locationRiskProfile	policy.setBasePrice(\$param);	System.out.println("\$param");
Base pricing rules	Age Bracket	Location risk prof	Base \$ AUD	Record Reason
Young safe package when	18, 24	LOW	450	
		MED	200	Priors not relevant
		MED	300	

```
Driver(age >= 18, age <= 24,  
       locationRiskProfile == "LOW",  
       priorClaims == "1")
```

```
policy:
```

```
Policy(type == "COMPREHENSIVE")
```

```
then
```

```
    policy.setBasePrice(450);
```

```
end
```

Decision Table

Age	< 18			18 to < 45			45 to < 60			>= 60		
Length of service	< 15	15 to < 30	>= 30	< 15	15 to < 30	>= 30	< 15	15 to < 30	>= 30	< 15	15 to < 30	>= 30
Assign 22 days	X	X	X	X	X	X	X	X	X	X	X	X
5 extra days	X	X	X			X			X	X	X	X
3 extra days					X		X	X				
2 extra days						X			X	X	X	X
	1	2	3	4	5	6	7	8	9	10	11	12

Expanded form

- Single column for every condition combination
- The number of columns should equal the product of the number of states for every condition.
 - e.g. 2 conditions, one with 3 states the other 4 (see above):
$$3 * 4 = 12 \text{ combinations}$$
 - e.g. 2 conditions each with 3 states and 1 condition with 4 states gives:
$$3 * 3 * 4 = 36 \text{ combinations}$$

Decision Table

Contracted form

- Contraction is the first optimisation.
- Reduces the number of condition columns.
- Removes impossible combinations
- If the same actions exist for rules covering all condition states for a given condition they can be combined and the condition state becomes irrelevant.

Decision Table

Contracted form – stage 1

Rules 2 and 3 are impossible conditions

Age	< 18			18 to < 45			45 to < 60			≥ 60		
Length of service	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30
Assign 22 days	X	X	X	X	X	X	X	X	X	X	X	X
5 extra days	X	X	X			X				X	X	X
3 extra days					X		X	X	X			
2 extra days						X				X	X	X
	1	2	3	4	5	6	7	8	9	10	11	12

Age	< 18			18 to < 45			45 to < 60			≥ 60			
Length of service	< 15	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30	< 15	15 to < 30	≥ 30
Assign 22 days	X	X	X	X	X	X	X	X	X	X	X	X	X
5 extra days	X				X					X	X	X	X
3 extra days				X		X	X	X					
2 extra days					X					X	X	X	X
	1	2	3	4	5	6	7	8	9	10	11	12	13

Decision Table

Contracted form – stage 2

Merge adjacent column groups with identical action parts

Age	< 18	18 to < 45			45 to < 60			>= 60		
Length of service	< 15	< 15	15 to < 30	>= 30	< 15	15 to < 30	>= 30	< 15	15 to < 30	>= 30
Assign 22 days	X	X	X	X	X	X	X	X	X	X
5 extra days	X			X				X	X	X
3 extra days			X		X	X	X			
2 extra days				X				X	X	X
	1	2	3	4	5	6	7	8	9	10

Age	< 18	18 to < 45			45 to < 60	>= 60		
Length of service	< 15	< 15	15 to < 30	>= 30	-	< 15	15 to < 30	>= 30
Assign 22 days	X	X	X	X	X	X	X	X
5 extra days	X			X		X	X	X
3 extra days			X		X			
2 extra days				X		X	X	X
	1	2	3	4	5	6	7	8

Redundancy - Subsumption

Age	-	-	< 18	>= 60	-	18 to < 60	45 to < 60	-	>= 60
Length of service	-	>= 30	-	-	>= 30	15 to < 30	< 30	>= 30	-
Employee of the Year	Y	Y	-	-	-	-	-	-	-
Assign 22 days									
5 extra days			X	X	X				
3 extra days						X	X		
2 extra days								X	X
1 extra day	X	X							
	1	2	3	4	5	6	7	8	9

Deficiency

- Deficiency
- Premium is £500 if applicant age is less than 30
- Premium is £300 if Years Without Claim is greater than or equal to 10 years.

Age	< 30	-
Years without claims	-	≥ 10
Premium £500	X	
Premium £300		X
	1	2

- Applicant is 29, premium is £500
- Applicant has 12 years without claim, premium is £300
- Applicant is 29 with 12 years without claim, premium is ?!?

Property Reactive

Loop Problem

```
rule "Salary award for min 2 years service" when
    e : Employee( lengthOfService > 2 )
then
    modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```

Loop Problem

```
rule "Salary award for min 2 years service" no-loop
when
    e : Employee( lengthOfService > 2 )
then
    modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```



Loop Problem

```
rule "Salary award for min 2 years service" no-loop when
  e : Employee( lengthOfService > 2 )
then
  modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```

```
rule "Salary award for min 8 years service" no-loop when
  e : Employee( lengthOfService > 8 )
then
  modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```



Loop Problem

```
rule "Salary award for min 2 years service" when
  e : Employee( lengthOfService > 2 )
    not SalaryMin2Years( employee == e )
```

then

```
  modify( e ) { setSalary( e.getSalary() * 1.05 ) };
  insert ( new SalaryMin2Years(e) );
```

end



```
rule "Salary award for min 8 years service" when
  e : Employee( lengthOfService > 8 )
    not SalaryMin8Years( employee == e )
```

then

```
  modify( e ) { setSalary( e.getSalary() * 1.05 ) };
  insert ( new SalaryMin8Years(e) );
```

end

Loop Problem

```
rule "Year End" when
    d : ChangeDate( )
    e : Employee( )
then
    modify( e ) { lengthOfService(
        d.getYear() - e.getStartYear() ) };
end
```

```
rule "Salary award for min 8 years service" when
    e : Employee( lengthOfService > 8 )
    not SalaryMin8Years( employee == e )
```

```
then
    modify( e ) { setSalary( e.getSalary() * 1.05 ) };
    insert ( new SalaryMin8Years(e) );
end
```



Refraction

This term comes from the neurobiological observation of a refractory period for a neuron, which means that the neuron is not able to fire immediately without first going through a relaxation process.

In a similar way, OPS5 will not allow the same instantiation in the conflict set from firing twice in a row. This prevents the inference engine from entering into an infinite loop.

W3C RIF Refraction

- Refraction

When a rule instance is fired, it is removed from the conflict set (and thus will not be created again even its condition part remains true), unless one of the objects in its condition part is modified again. In the later case, the repeatability concept determines how the rule instance is treated

W3C RIF Refraction

- Repeatability

After the execution of a rule instance, the rule instance is removed from the conflict set by the refraction. Later on, if one of the objects in the condition part is modified and if the rule evaluates to true, the repeatability controls whether if the rule instance can be created again.

Loop Problem (Refraction)

rule "Salary award for min 2 years service"

 repeatable false **when**

 e : Employee(lengthOfService > 2)

then

 modify(e) { setSalary(e.getSalary() * 1.05) };

end

rule "Salary award for min 8 years service"

 repeatable false **when**

 e : Employee(lengthOfService > 8)

then

 modify(e) { setSalary(e.getSalary() * 1.05) };

end



AAAAAAahhhhhhhh



FRUSTRATION

sometimes a punch to the face is needed to get someones attention



KEEP
CALM
AND
CARRY
ON

Annotate Class

```
@PropertyReactive  
public class Employee {  
    int salary;  
    int lengthOfService  
  
    // getters and setters below  
}
```

Loop Problem Fixed

```
rule "Salary award for min 2 years service" when
    e : Employee( lengthOfService > 2 )
then
    modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```

```
rule "Salary award for min 8 years service" when
    e : Employee( lengthOfService > 8 )
then
    modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```

@Watch

```
rule "Salary award for min 2 years service" when
  e : Employee( salary < 1000, lengthOfService > 2 )
    @Watch( !salary )
then
  modify( e ) { setSalary( e.getSalary() * 1.05 ) };
end
```

@Watch

```
rule "Record Salary Changes" when
    e : Employee( ) @Watch( salary )
then
    insert( new SalaryChange( e, e.getSalary() ) );
end
```

@Watch

@Watch(salary, lengthOfService, age)

@Watch(*)

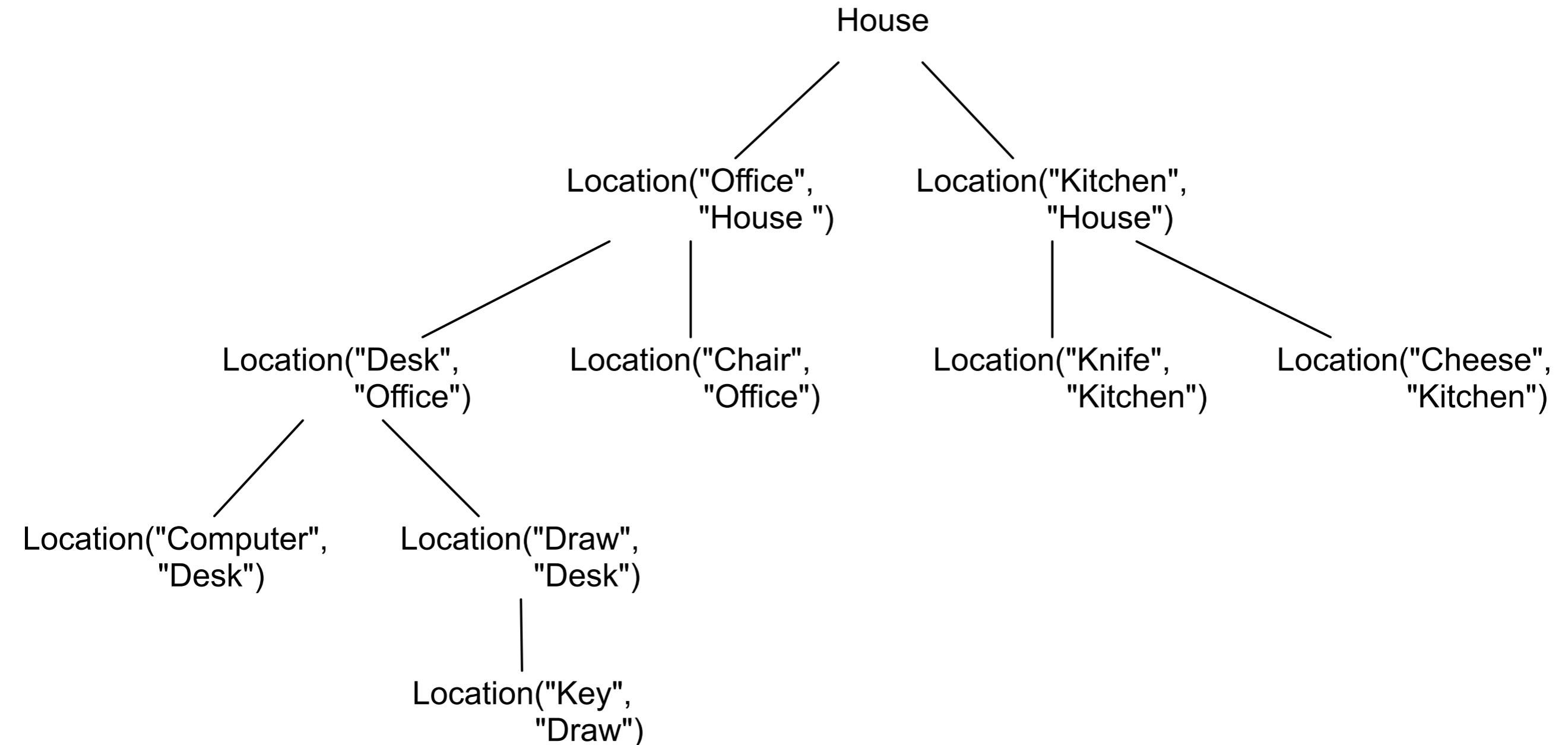
@Watch(!*)

@Watch(*, !salary)

@Watch(!*, salary)

Backward Chaining

Reasoning with Graphs

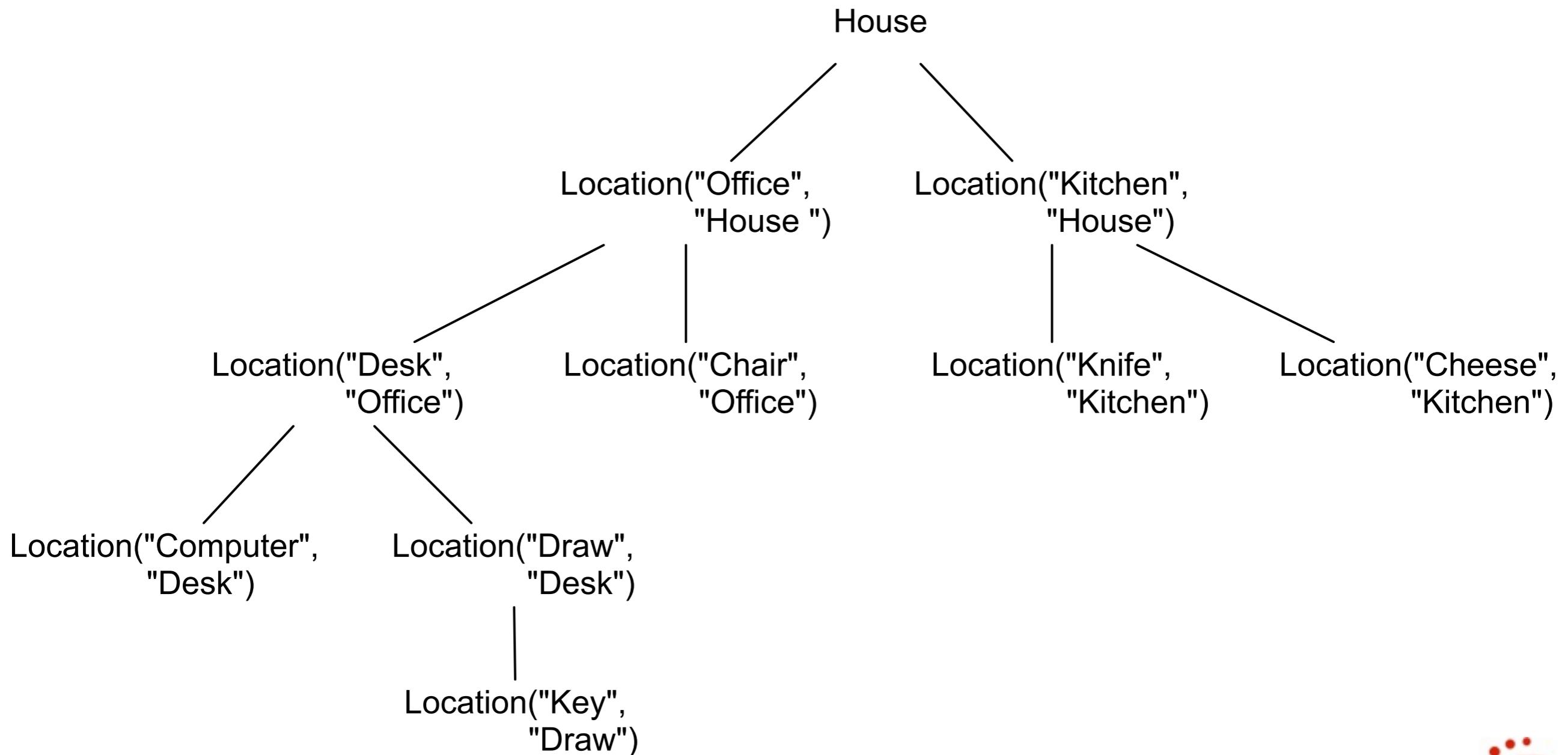


Backward Chaining

```

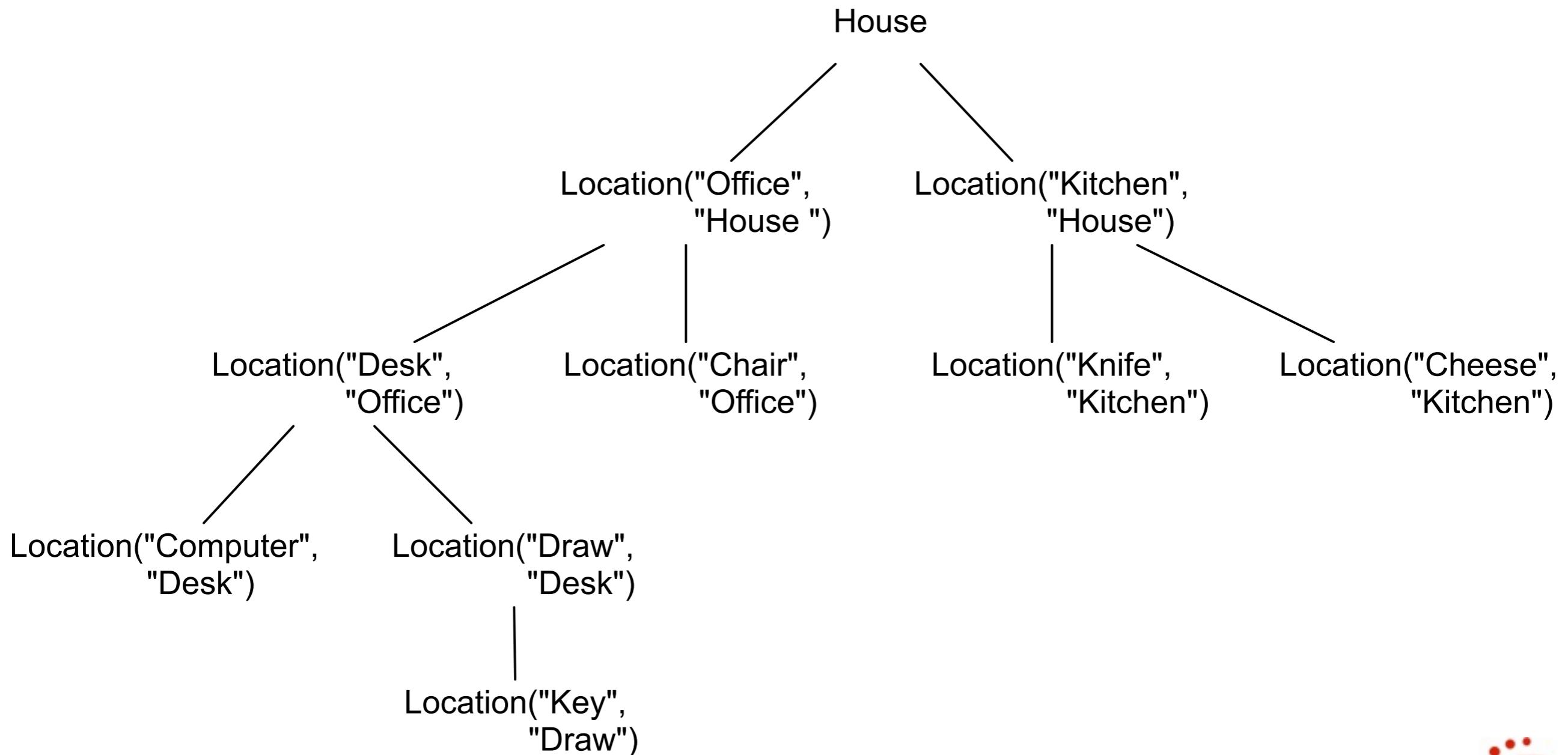
query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



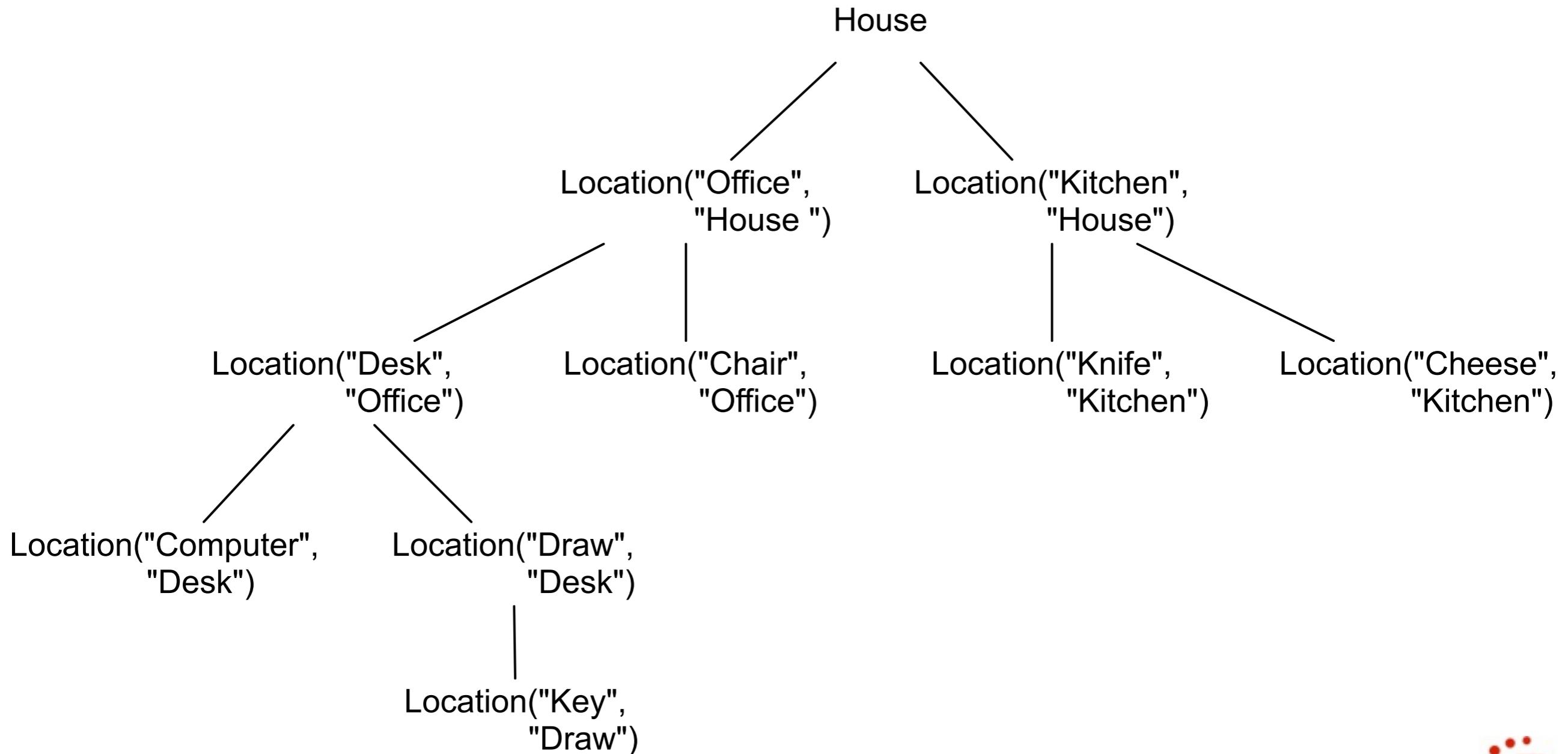
Backward Chaining

```
ksession.insert( new Location("Office", "House") );
ksession.insert( new Location("Kitchen", "House") );
ksession.insert( new Location("Knife", "Kitchen") );
ksession.insert( new Location("Cheese", "Kitchen") );
ksession.insert( new Location("Desk", "Office") );
ksession.insert( new Location("Chair", "Office") );
ksession.insert( new Location("Computer", "Desk") );
ksession.insert( new Location("Draw", "Desk") );
```



Backward Chaining

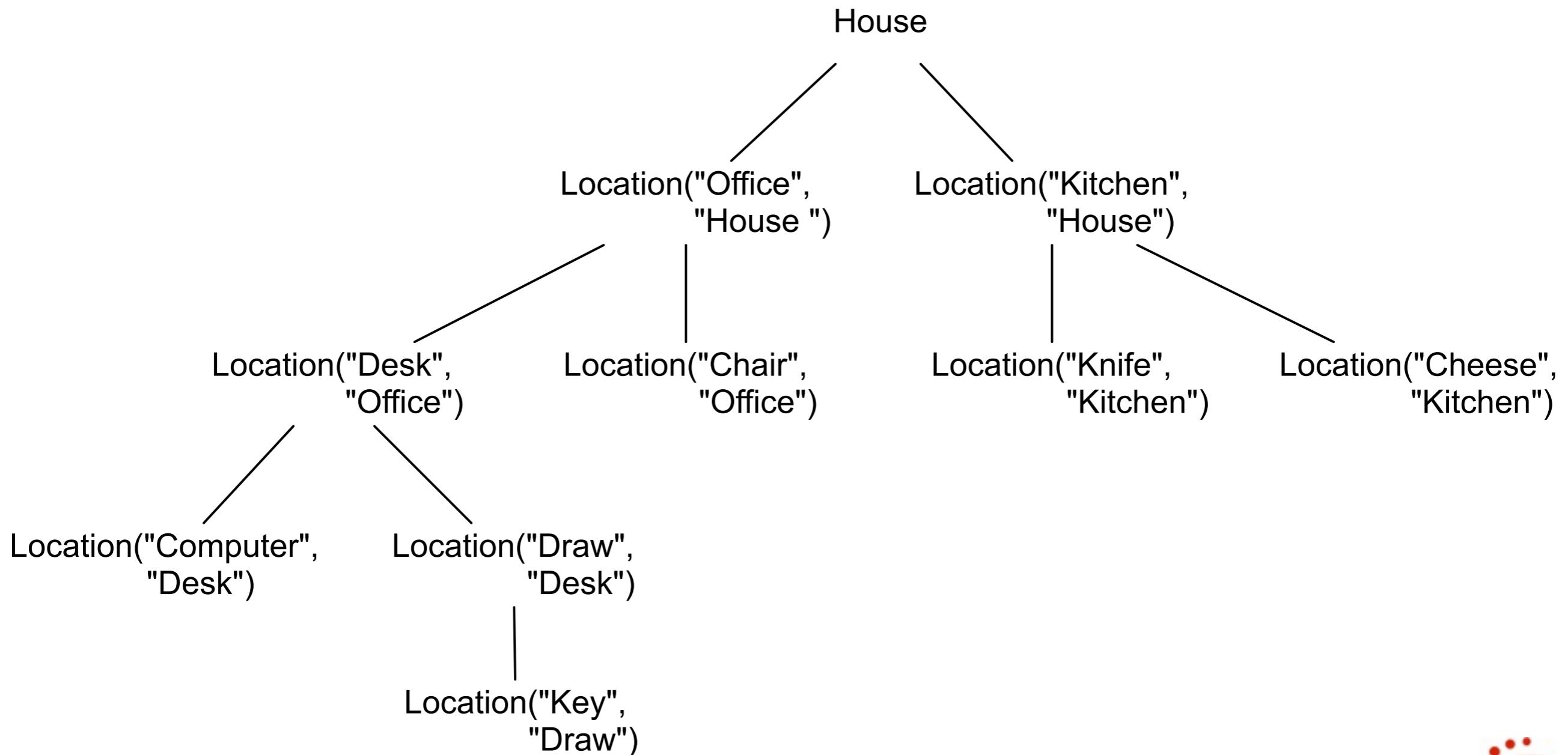
```
rule "go" salience 10
when
  $s : String( )
then
  System.out.println( $s );
end
```



Backward Chaining

```
rule "go" salience 10
when
  $s : String( )
then
  System.out.println( $s );
end
```

```
rule "go1"
when
  String( this == "go1" )
  isContainedIn("Office", "House"; )
then
  System.out.println( "office is in the house" );
end
```



Backward Chaining

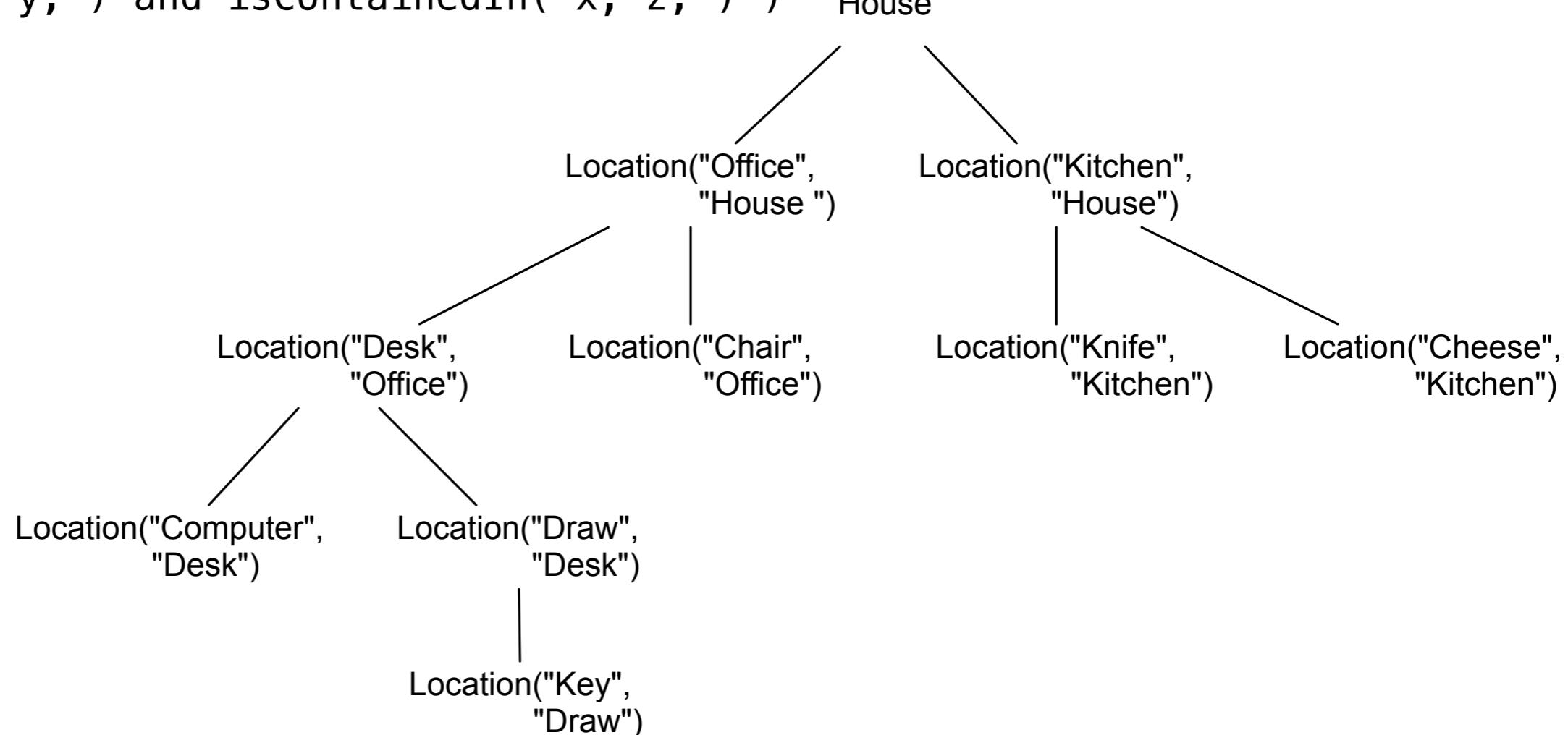
```

rule "go" salience 10
when
  $s : String(  )
then
  System.out.println( $s );
end

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end
  
```

```

rule "go1"
when
  String( this == "go1" )
  isContainedIn("Office", "House"; )
then
  System.out.println( "office is in the house" );
end
  
```



Backward Chaining

```

rule "go" salience 10
when
  $s : String( )
then
  System.out.println( $s );
end

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

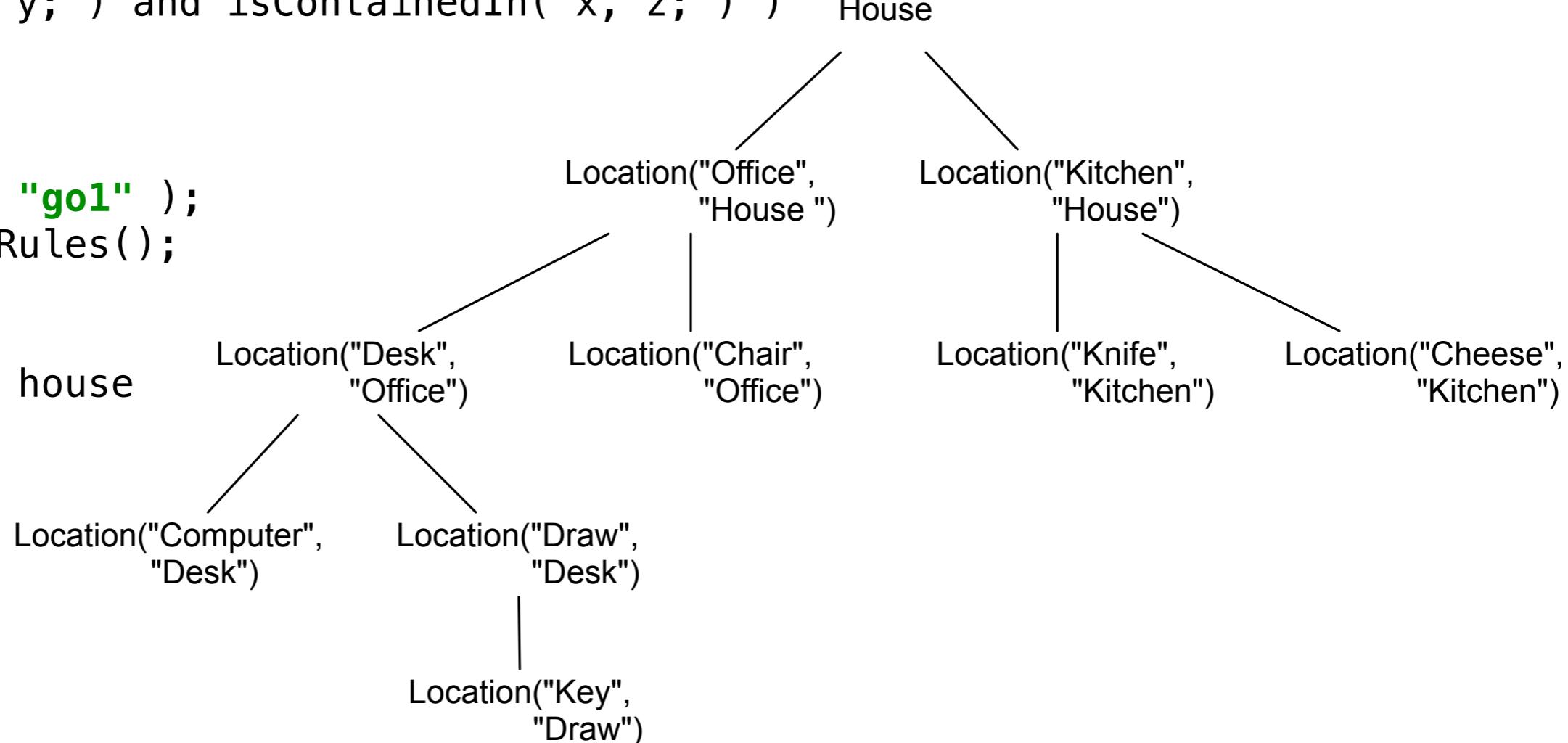
ksession.insert( "go1" );
ksession.fireAllRules();
---
go1
office is in the house

```

```

rule "go1"
when
  String( this == "go1" )
  isContainedIn("Office", "House"; )
then
  System.out.println( "office is in the house" );
end

```



Backward Chaining

```

rule "go" salience 10
when
  $s : String( )
then
  System.out.println( $s );
end

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

ksession.insert( "go1" );
ksession.fireAllRules();
---
go1
office is in the house

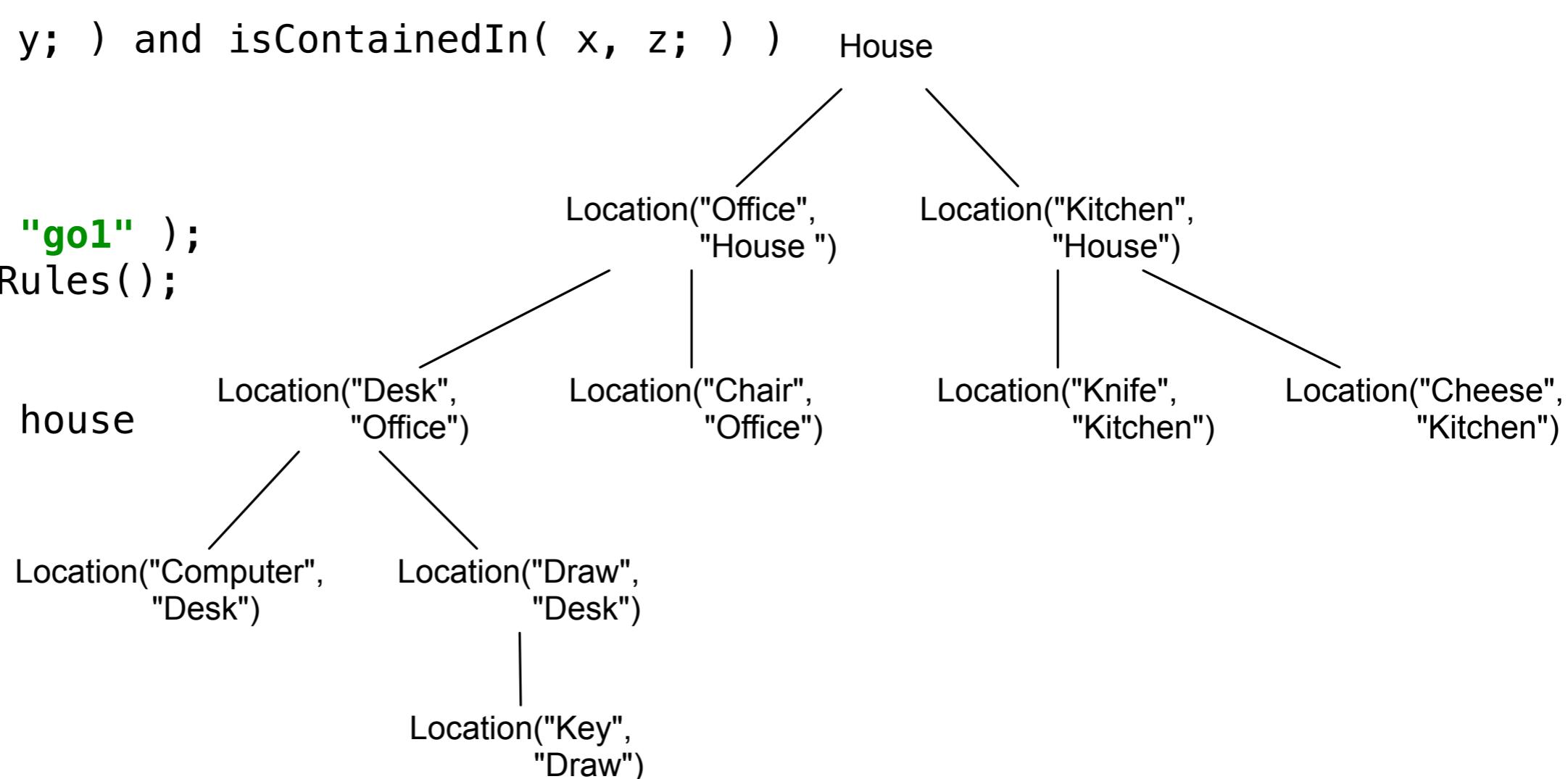
```

```

rule "go1"
when
  String( this == "go1" )
  isContainedIn("Office", "House"; )
then
  System.out.println( "office is in the house" );
end

```

isContainedIn(x==Office, y==House)



Backward Chaining

```

rule "go" salience 10
when
  $s : String( )
then
  System.out.println( $s );
end

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end
  
```

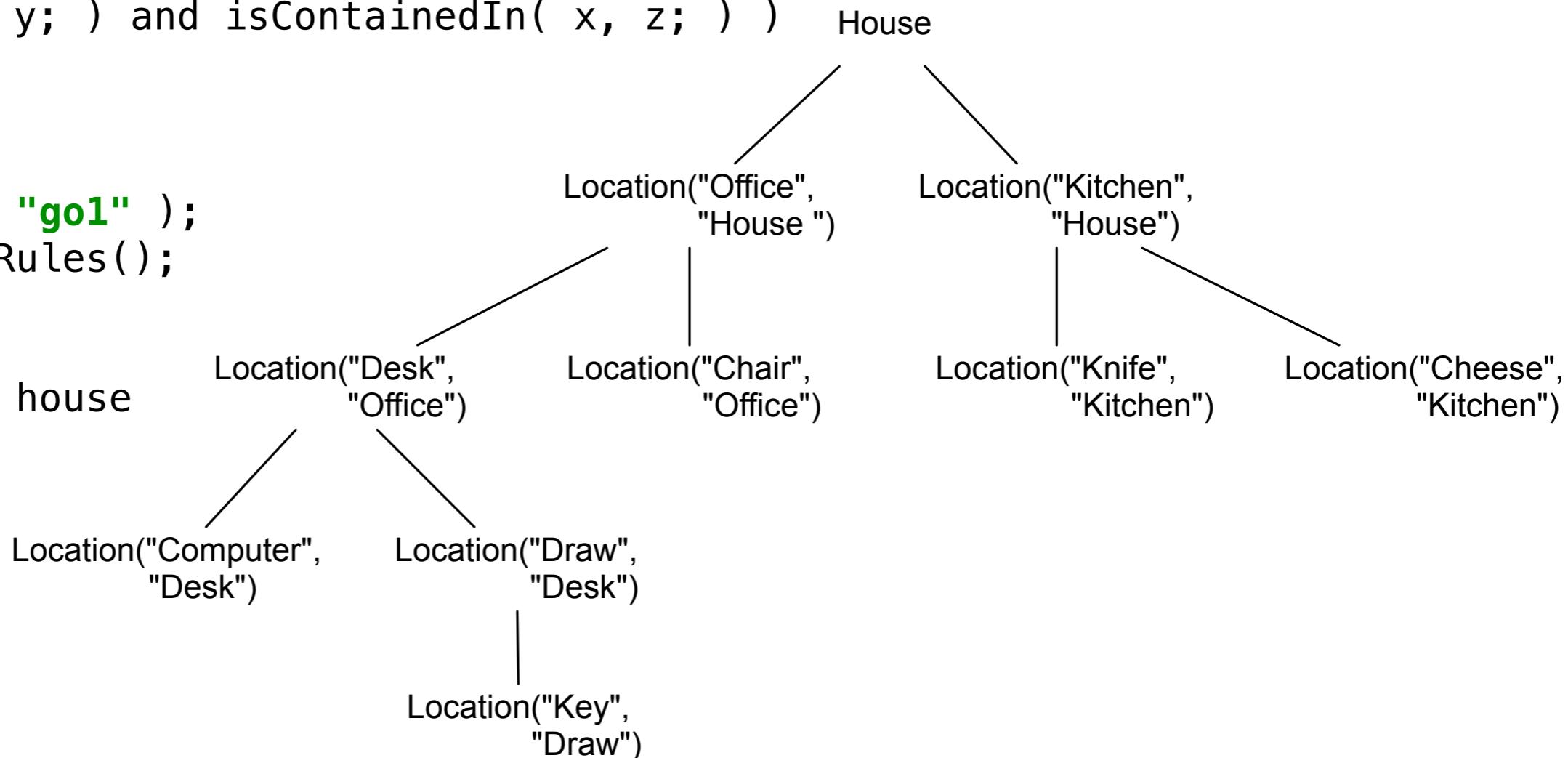
```

ksession.insert( "go1" );
ksession.fireAllRules();
---
go1
office is in the house
  
```

```

rule "go1"
when
  String( this == "go1" )
  isContainedIn("Office", "House"; )
then
  System.out.println( "office is in the house" );
end

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  isContainedIn(x==Office, y==House)
  Location(x==Office, y==House)
end
  
```

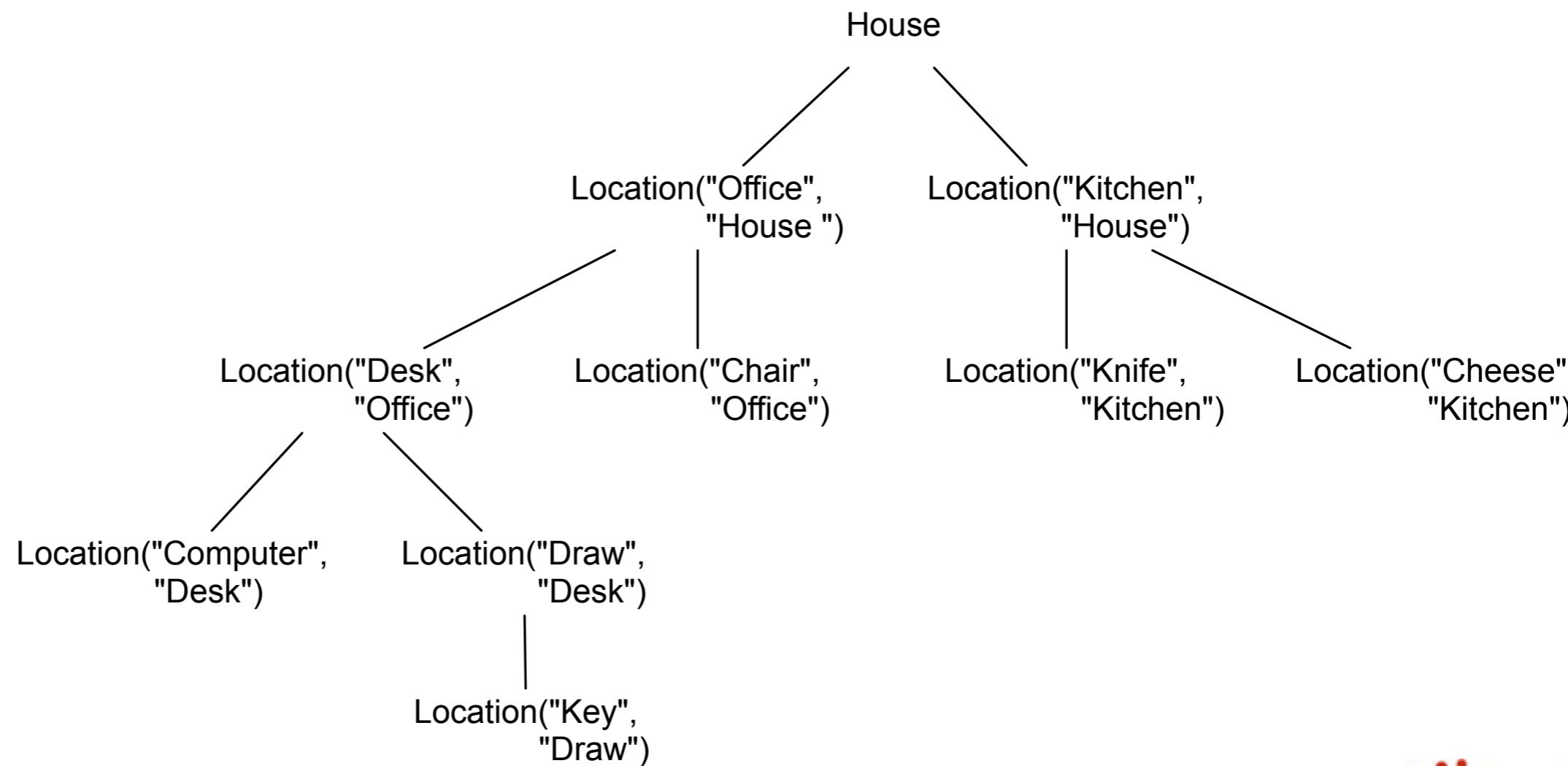


Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

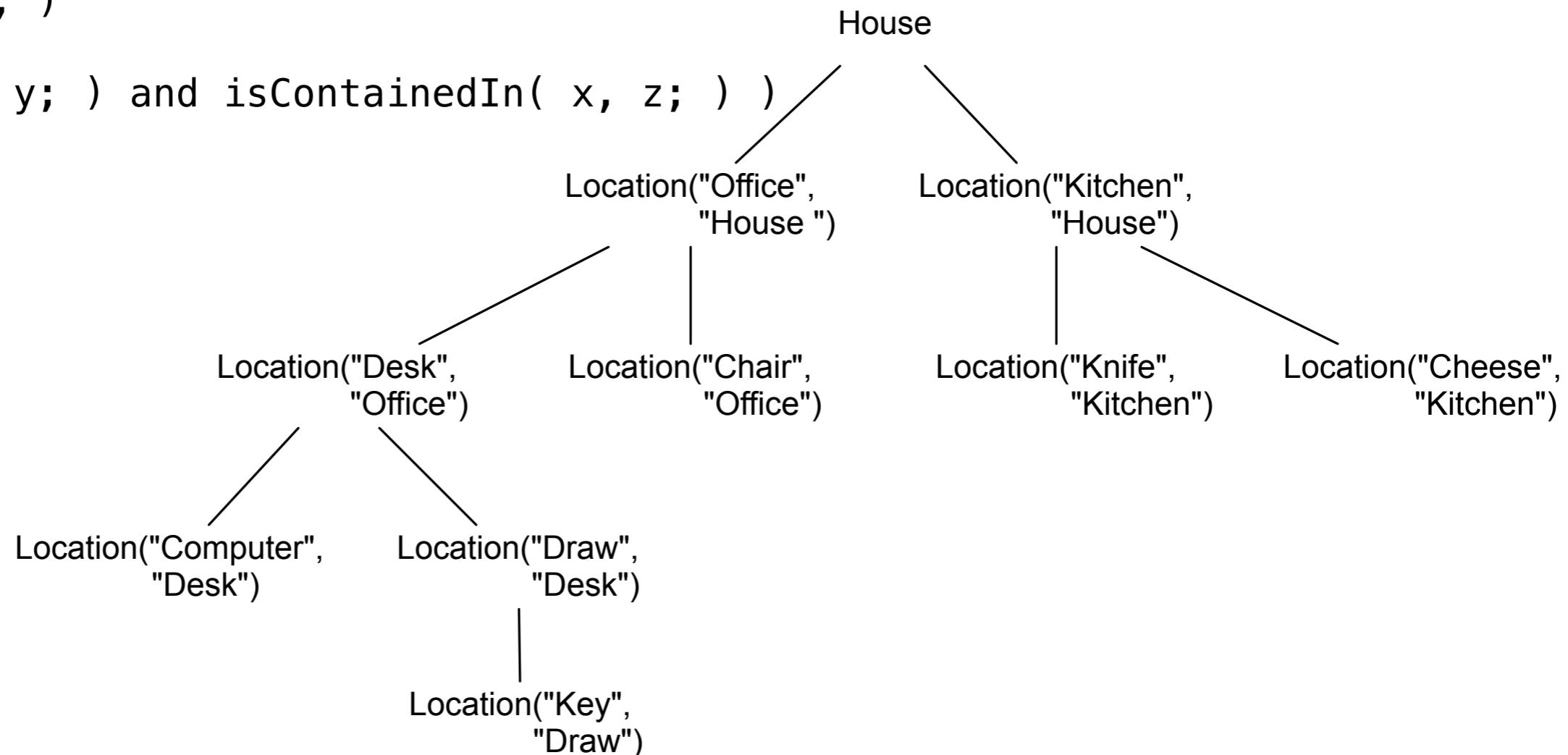
```



Backward Chaining

```
rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end
```

```
query isContainedIn( String x, String y )
Location( x, y; )
or
( Location( z, y; ) and isContainedIn( x, z; ) )
end
```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

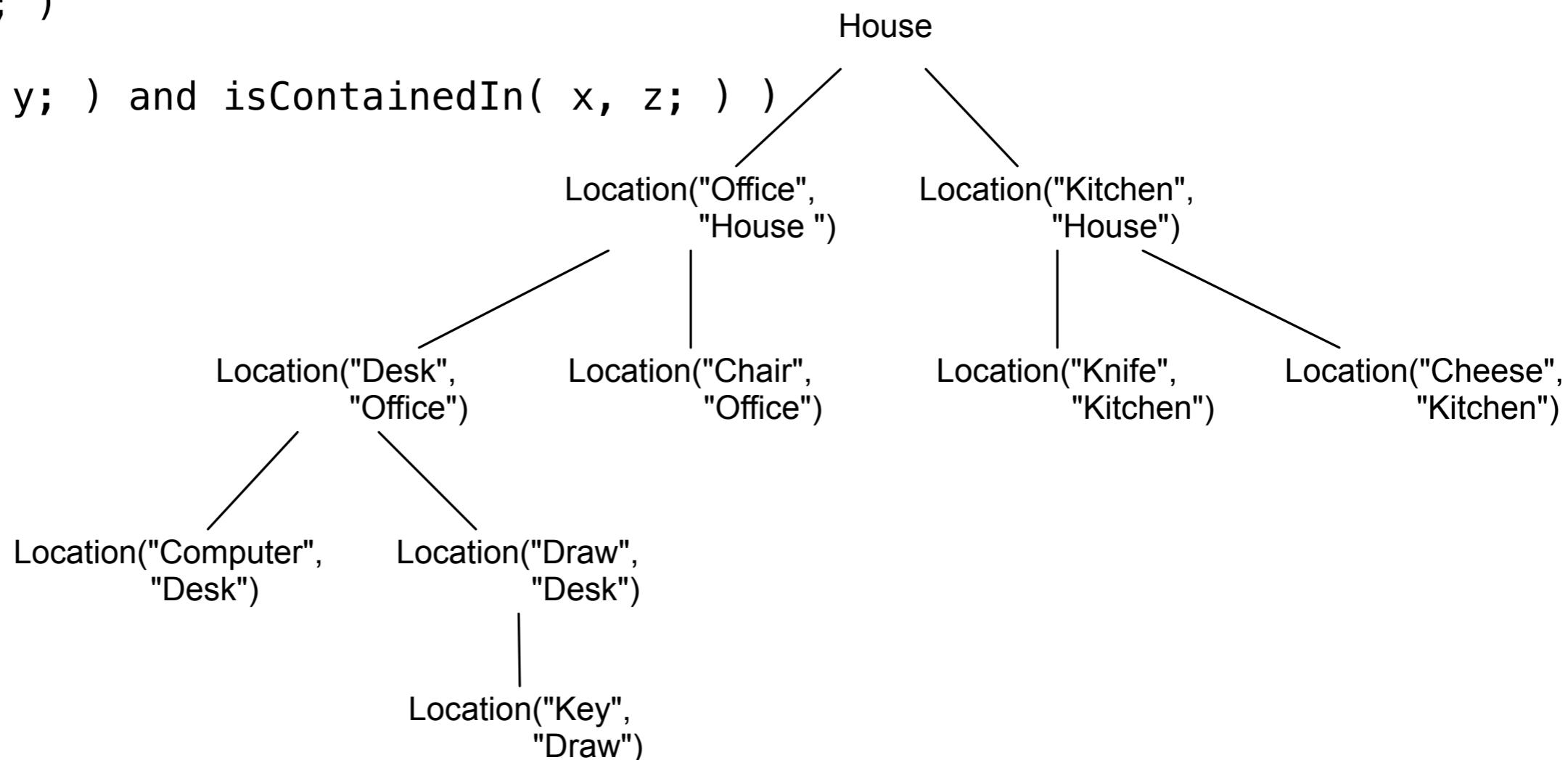
ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

```

query isContainedIn( String x, String y )
Location( x, y; )
or
( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( Draw in the House );
end

```

```

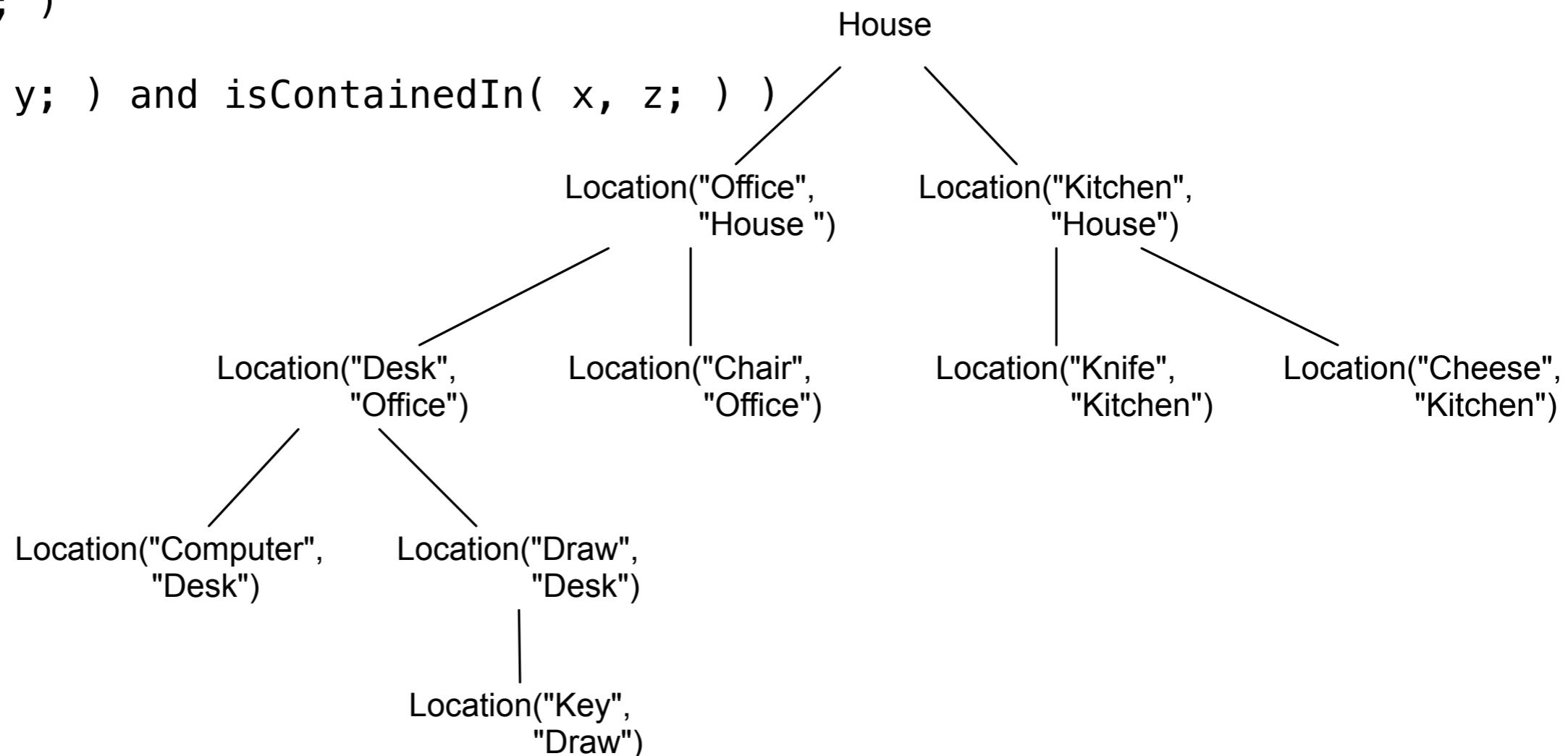
ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House
isContainedIn(x==Draw, y==House)

```

```

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

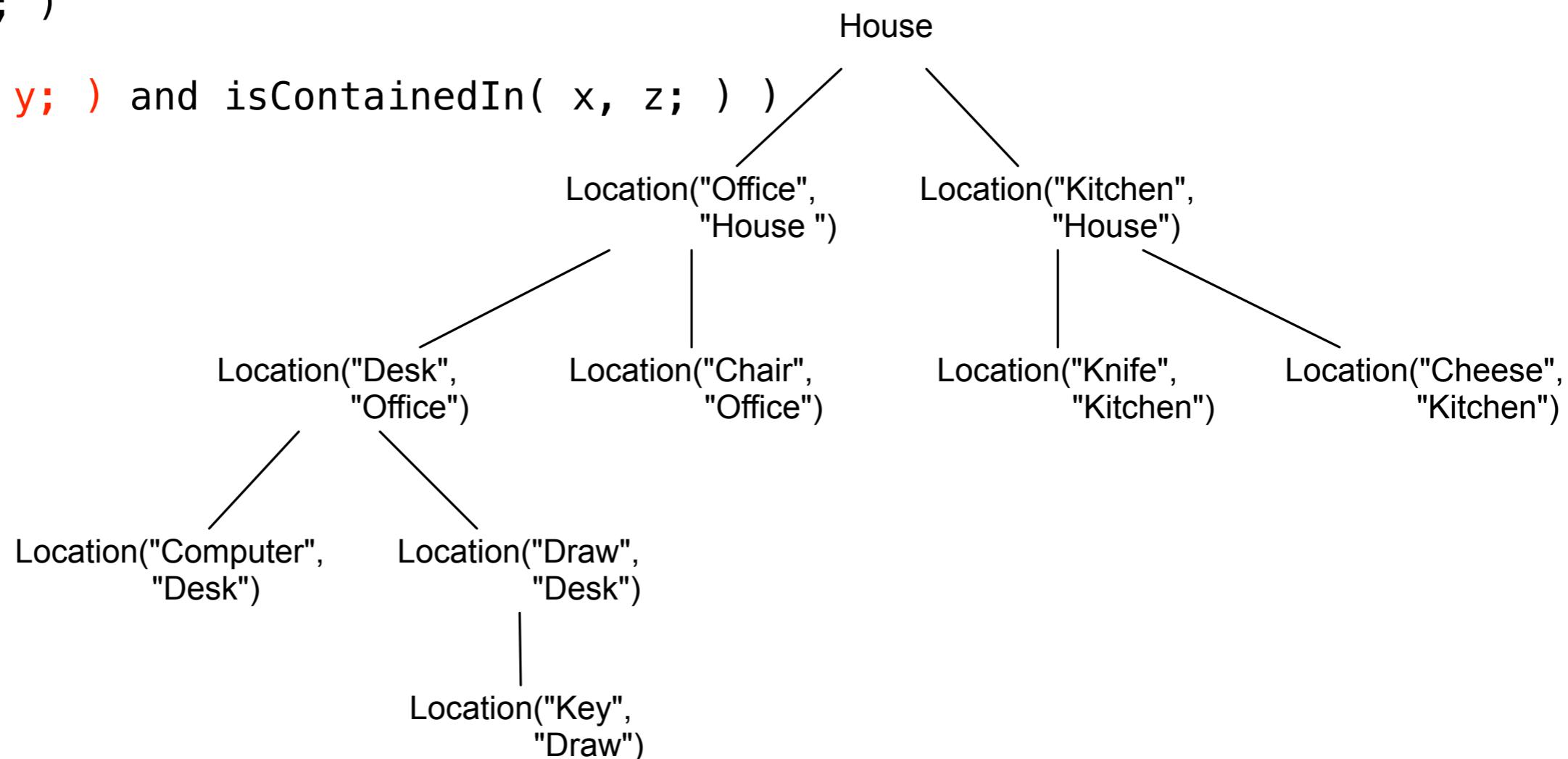
query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House
isContainedIn(x==Draw, y==House)
Location(z==Office, y==House)

```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( Draw in the House );
end

```

```

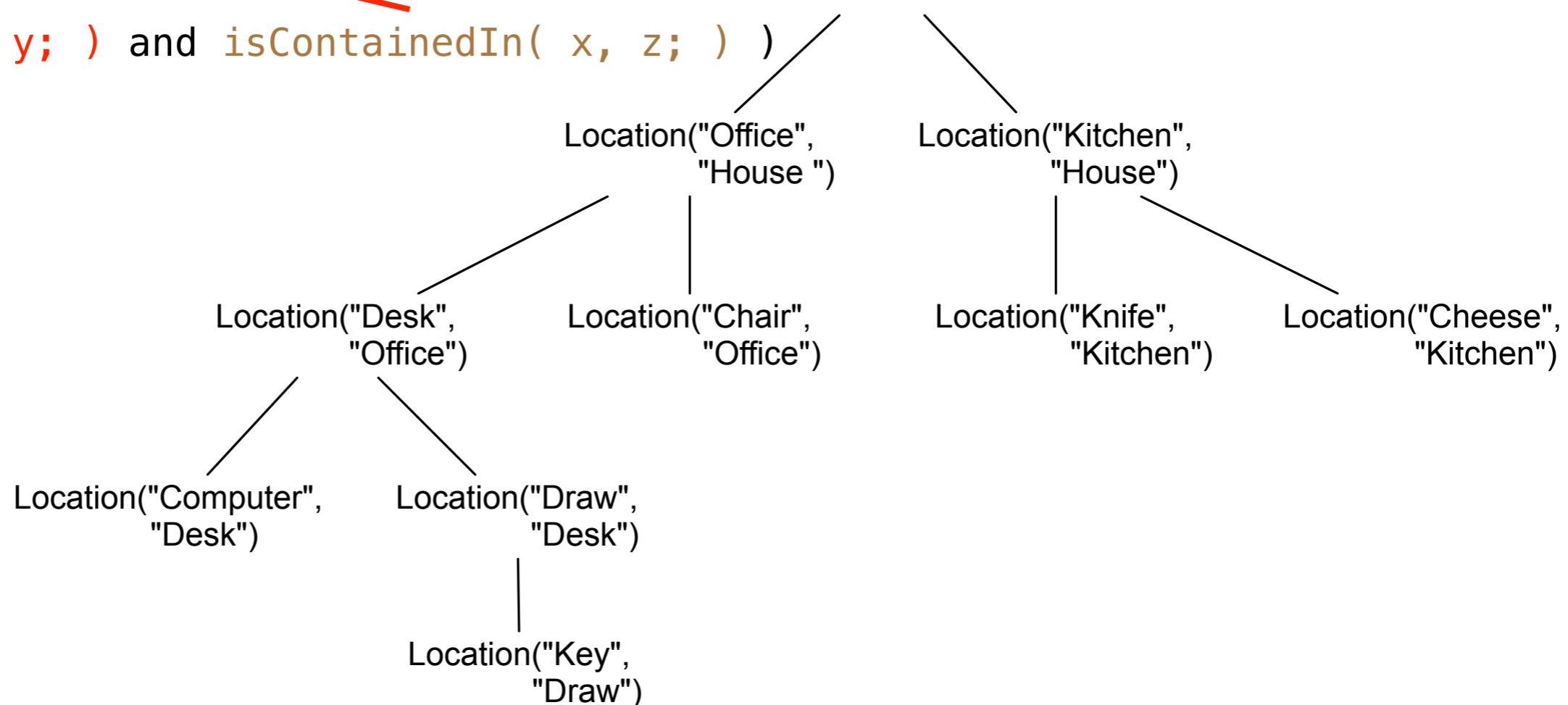
query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House
isContainedIn(x==Draw, y==House)
Location(z==Office, y==House)
isContainedIn(x==Draw, z==Office)

```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

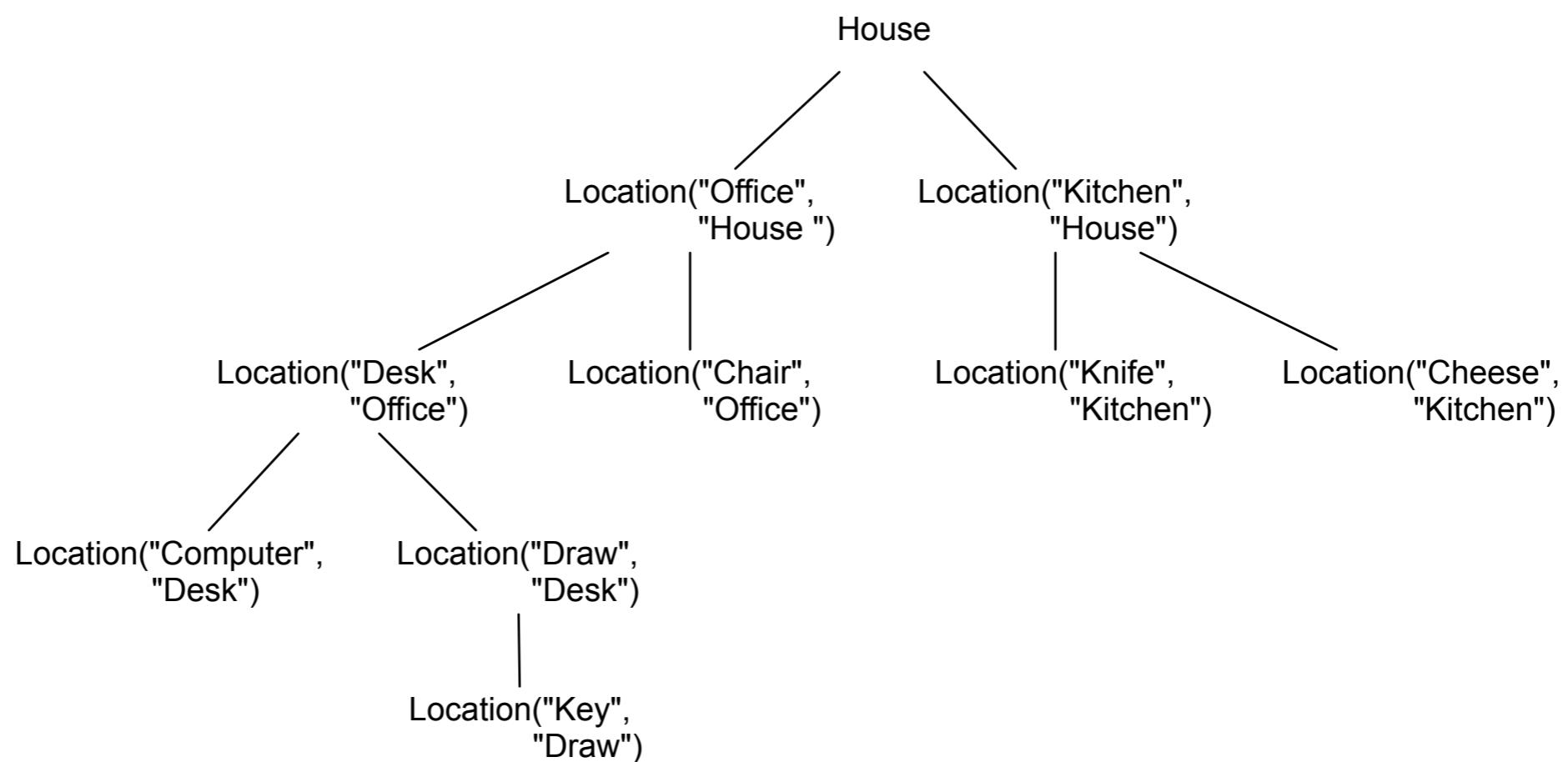
```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

isContainedIn(x==Draw, y==House)
Location(z==Office, y==House)
isContainedIn(x==Draw, z==Office)
Location(z==Kitchen, y==House)
isContainedIn(x==Draw, z==Kitchen)

```



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( Draw in the House );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

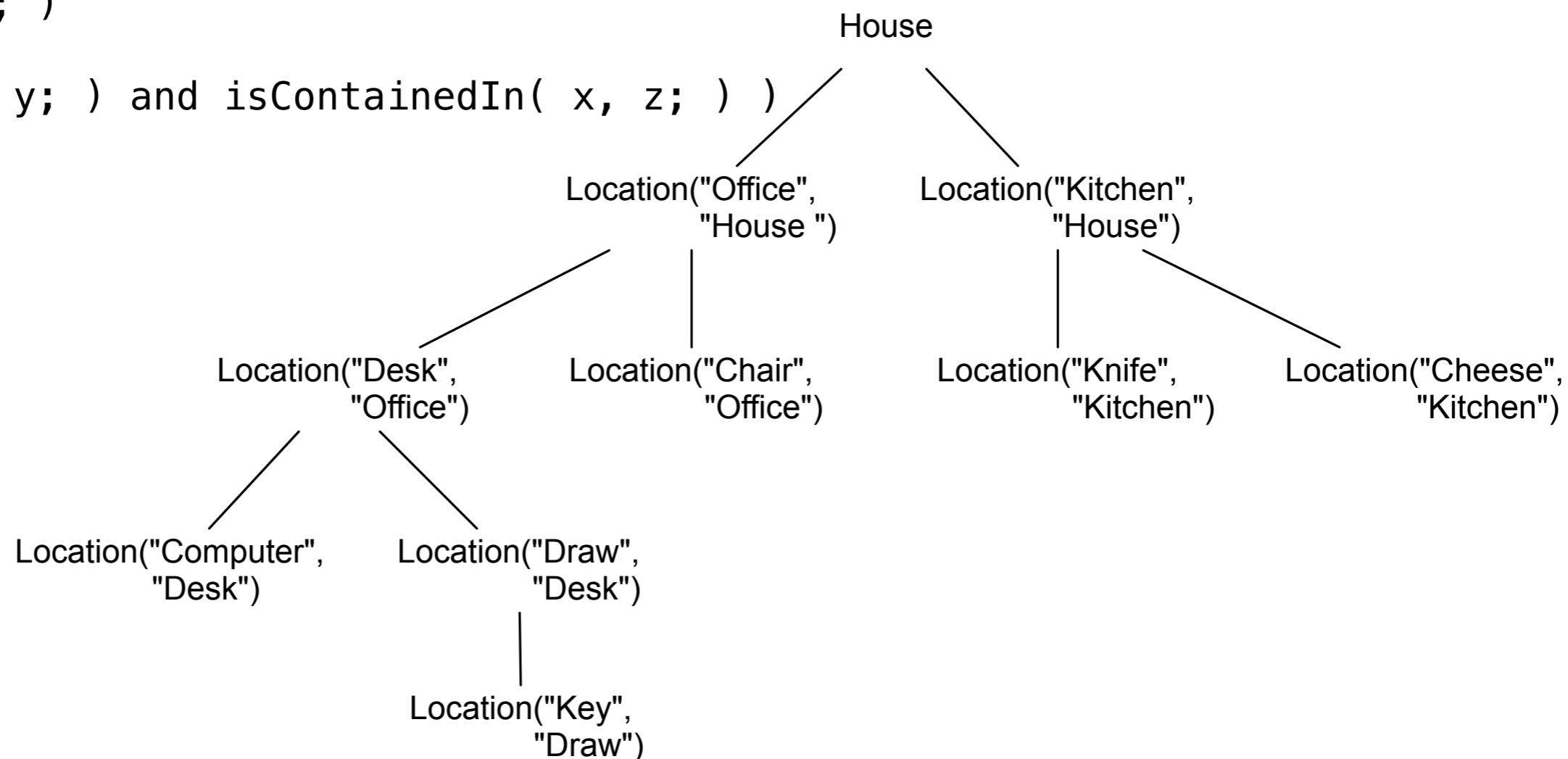
```

```

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Office)



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

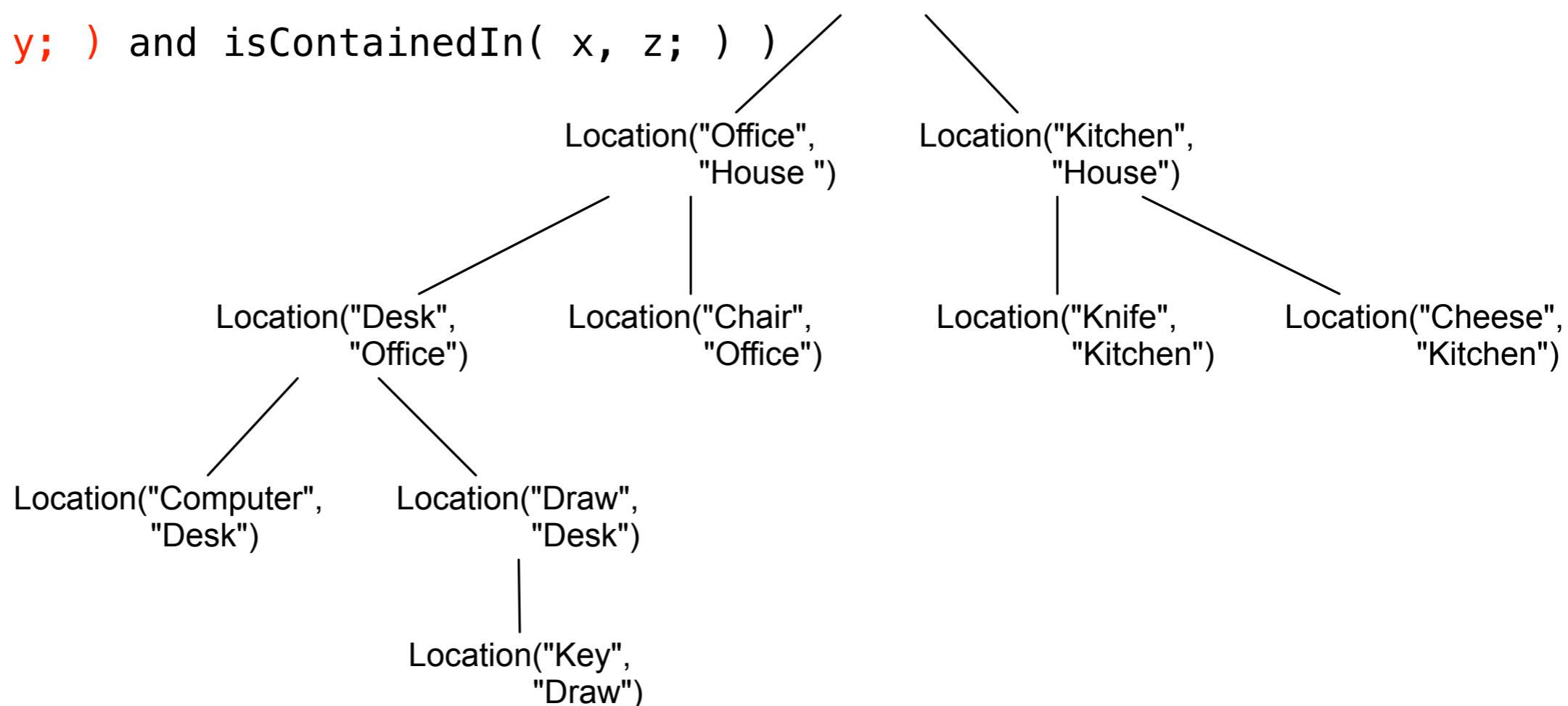
```

query isContainedIn( String x, String y )
Location( x, y; )
or
( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Office)
 Location(z==Desk, y==Office)

House



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

query isContainedIn( String x, String y )
Location( x, y; )
or
( Location( z, y; ) and isContainedIn( x, z; ) )
end

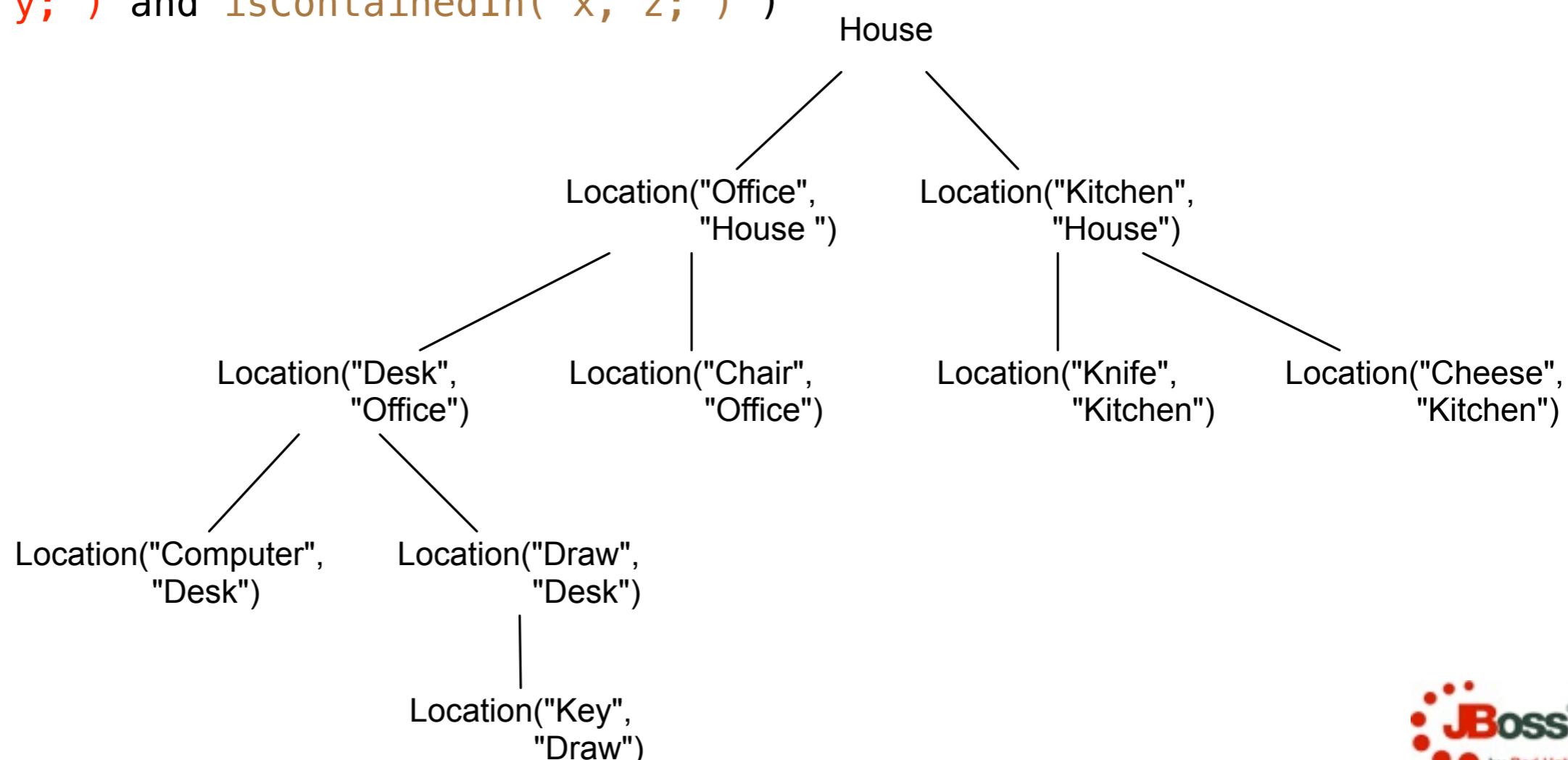
```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

isContainedIn(x==Draw, y==Office)
 Location(z==Desk, y==Office)
 isContainedIn(x==Draw, z==Desk)



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

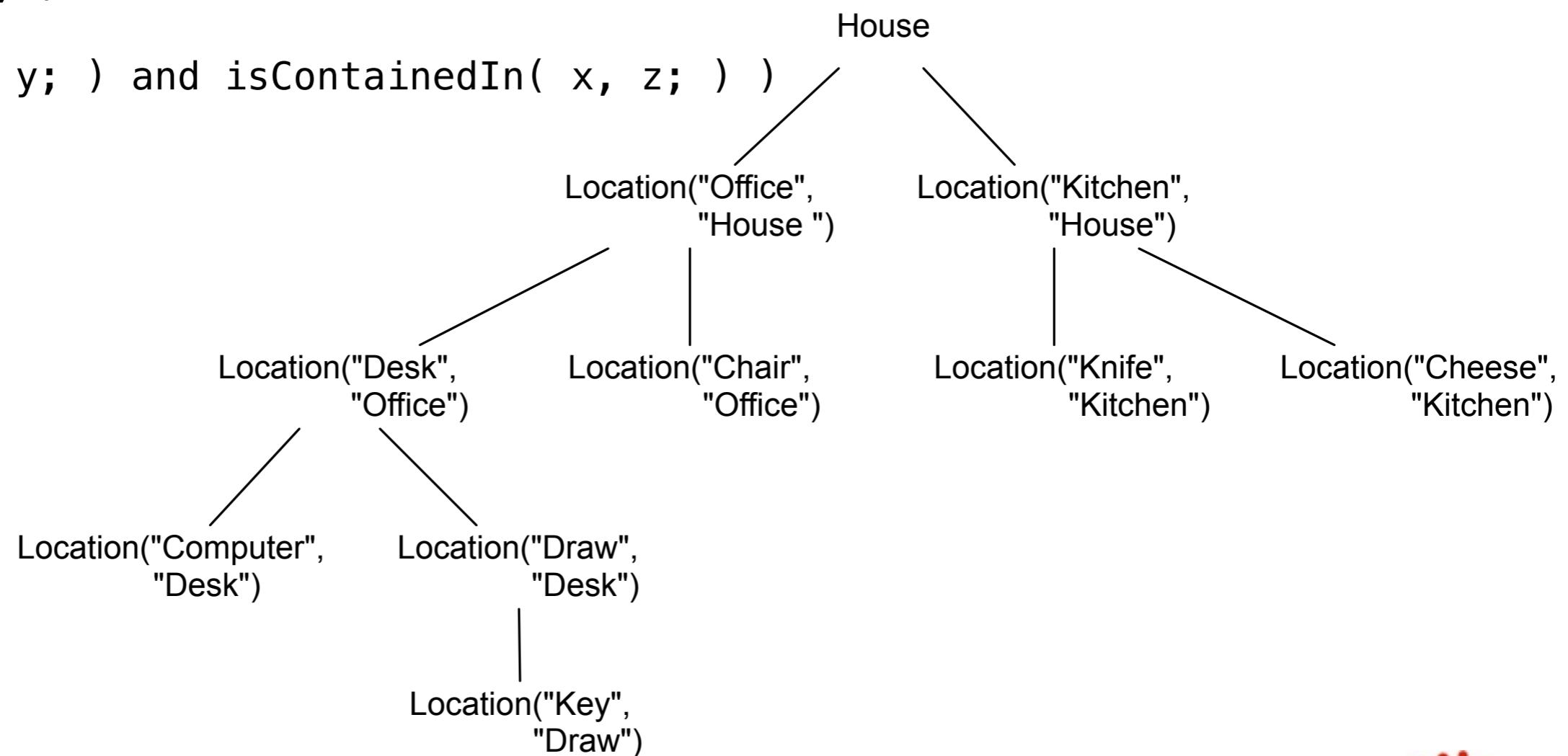
```

```

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Desk)



Backward Chaining

```

rule "go2"
when
  String( this == "go2" )
  isContainedIn("Draw", "House"; )
then
  System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

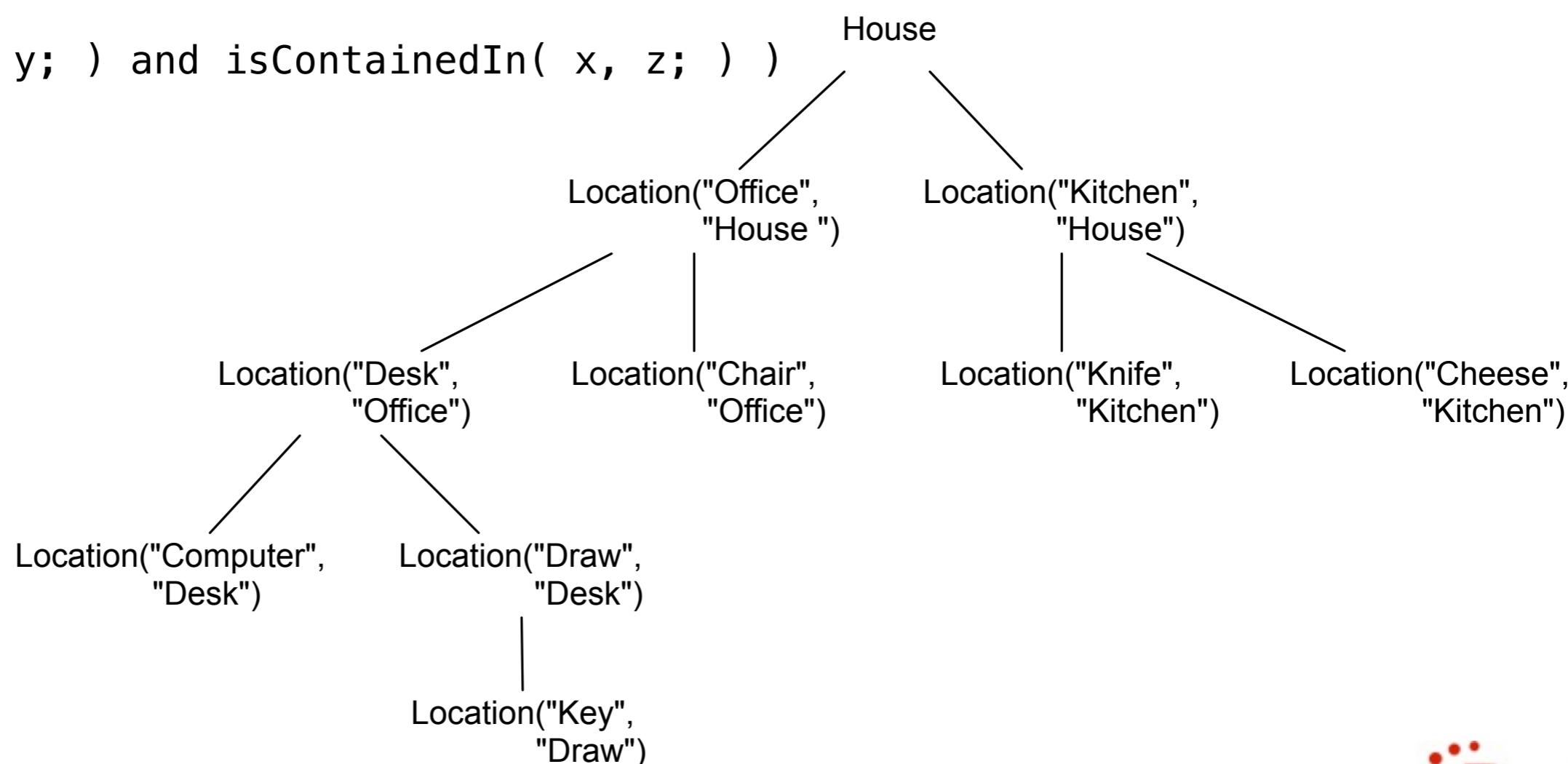
```

query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Desk)

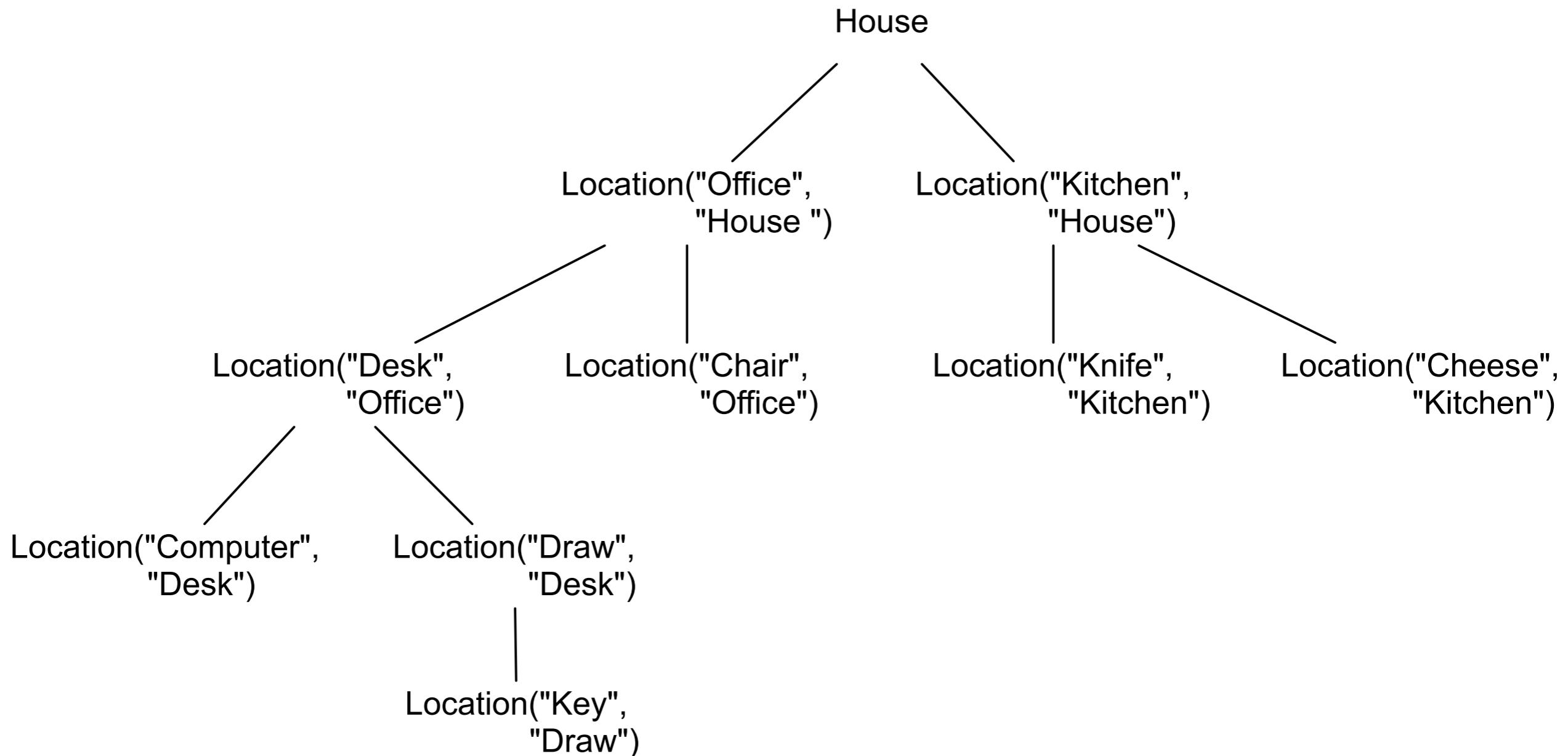
Location(x==Draw, y==Desk)



Backward Chaining

```

rule "go3"
when
  String( this == "go3" )
  isContainedIn("Key", "Office"; )
then
  System.out.println( "Key in the Office" );
end
  
```



Backward Chaining

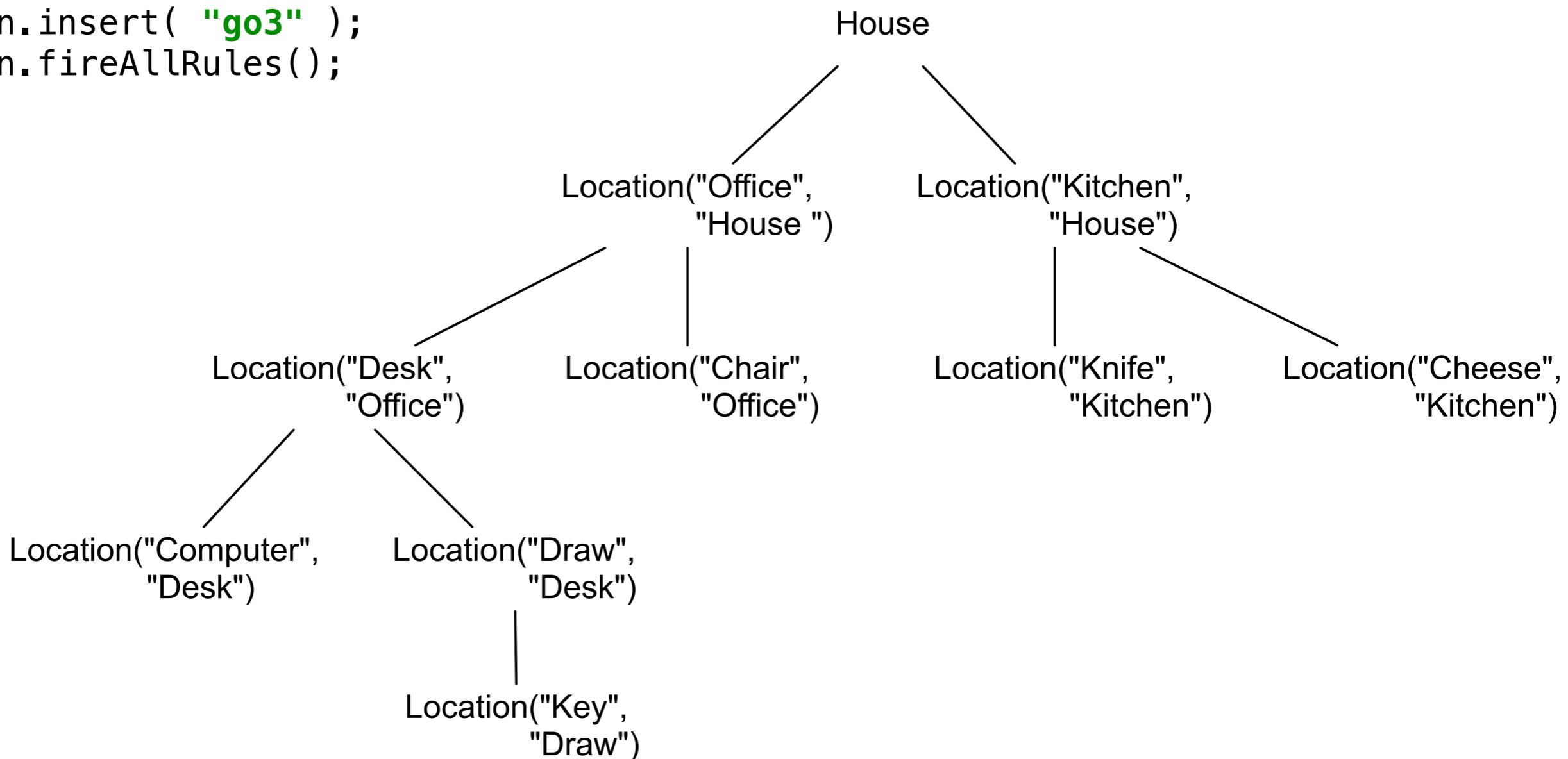
```

rule "go3"
when
  String( this == "go3" )
  isContainedIn("Key", "Office"; )
then
  System.out.println( "Key in the Office" );
end

```

```

ksession.insert( "go3" );
ksession.fireAllRules();
---
```



Backward Chaining

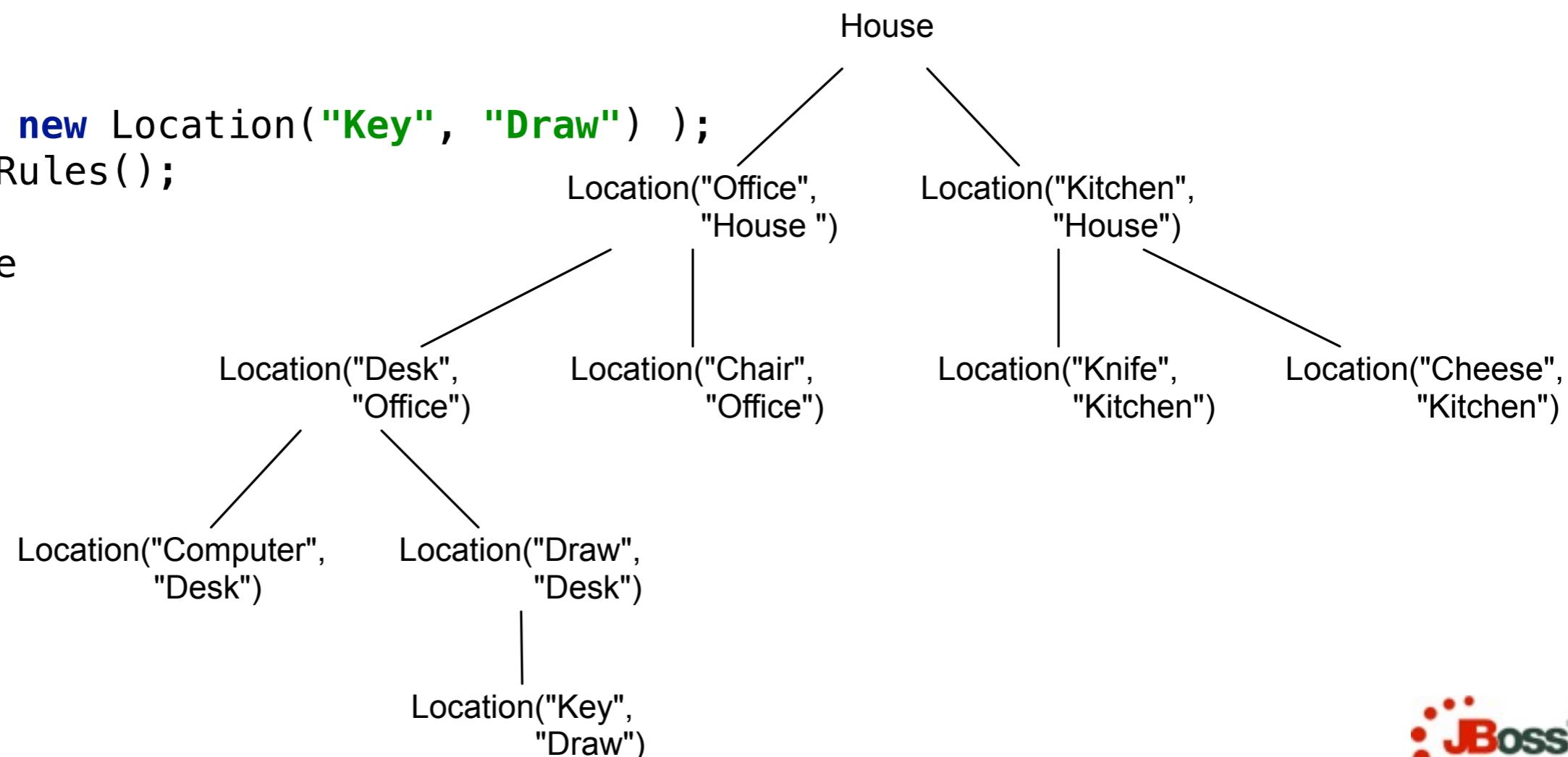
```
rule "go3"
when
  String( this == "go3" )
  isContainedIn("Key", "Office"; )
then
  System.out.println( "Key in the Office" );
end
```

```
ksession.insert( "go3" );
ksession.fireAllRules();
---
```

go3

```
ksession.insert( new Location("Key", "Draw") );
ksession.fireAllRules();
---
```

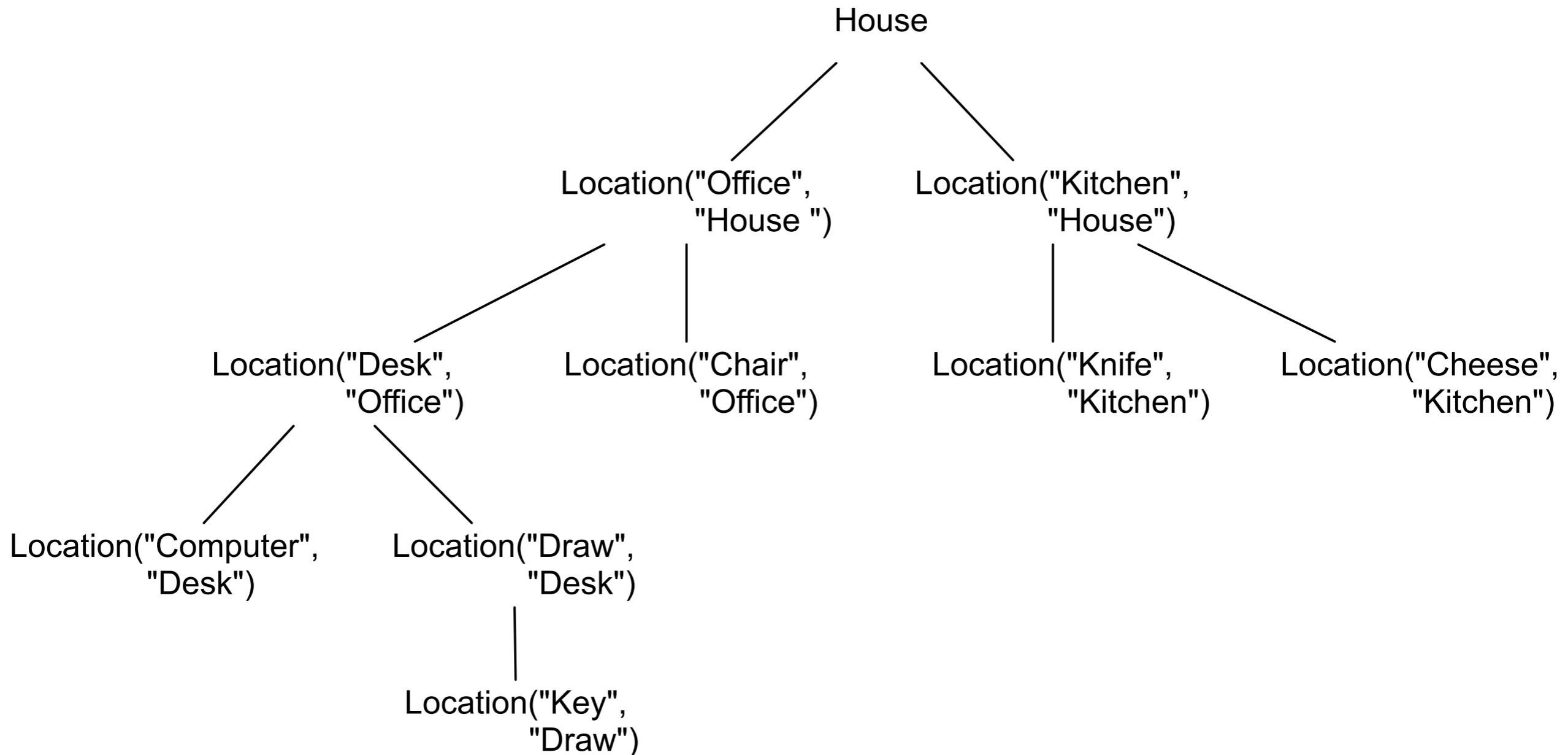
Key in the Office



Backward Chaining

```

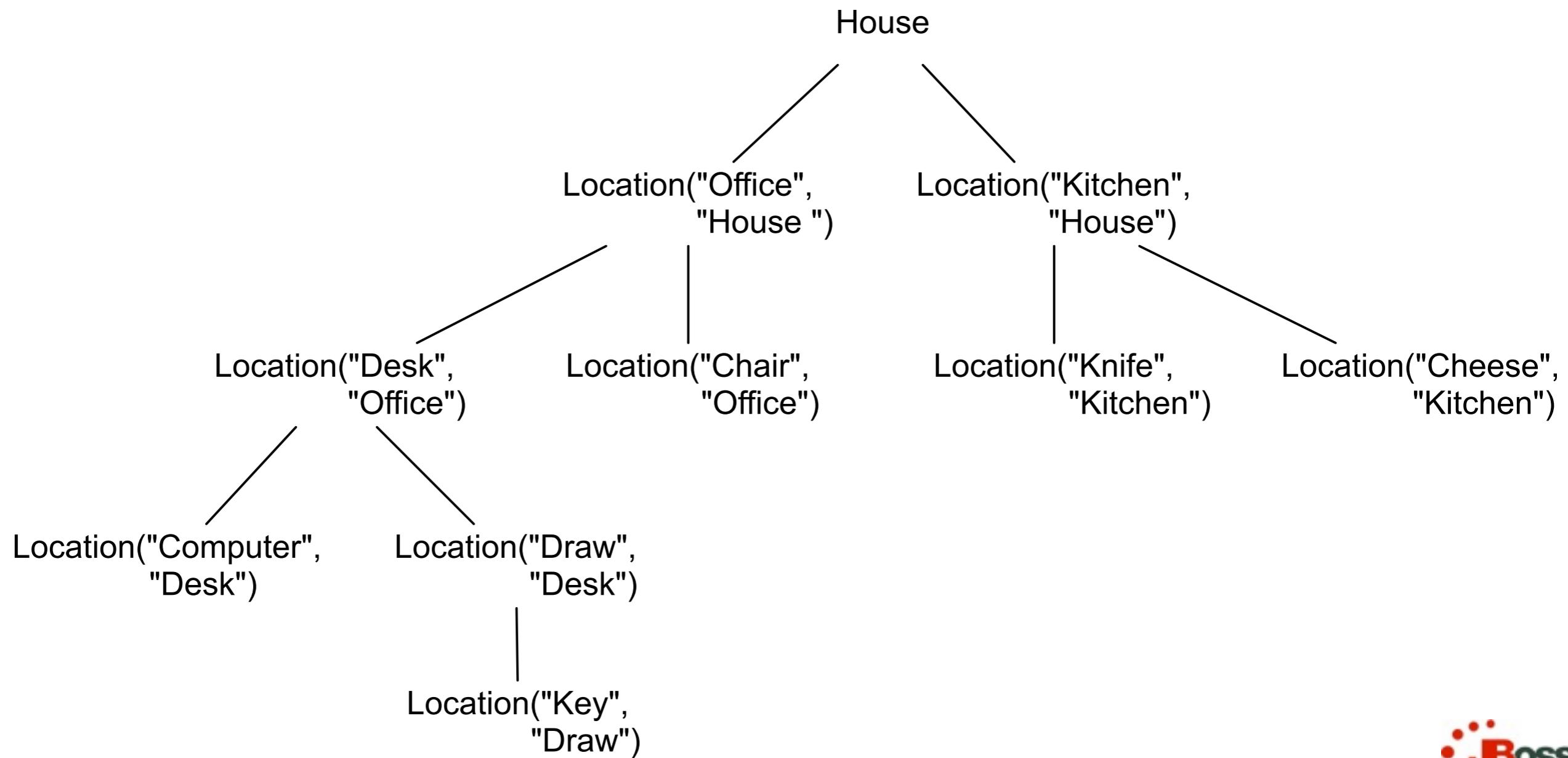
rule "go4"
when
  String( this == "go4" )
  isContainedIn(thing, "Office"; )
then
  System.out.println( "thing " + thing + " is in the Office" );
end
  
```



Backward Chaining

```

rule "go4"
when
  String( this == "go4" )
  isContainedIn(thing, "Office"; )
then
  System.out.println( "thing " + thing + " is in the Office" );
end
  
```



Backward Chaining

```

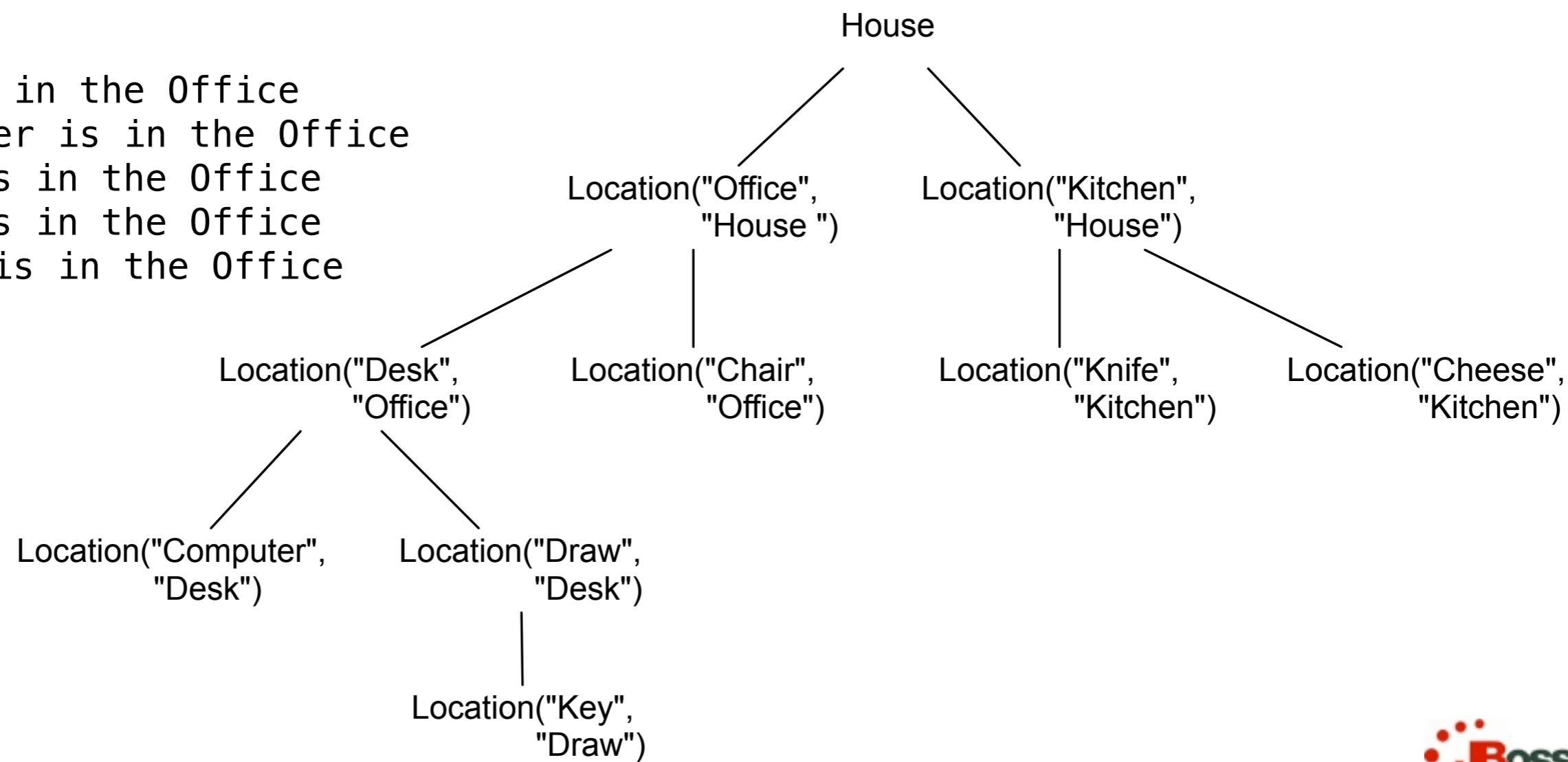
rule "go4"
when
  String( this == "go4" )
  isContainedIn(thing, "Office"; )
then
  System.out.println( "thing " + thing + " is in the Office" );
end
  
```

```

ksession.insert( "go4" );
ksession.fireAllRules();
---
```

```

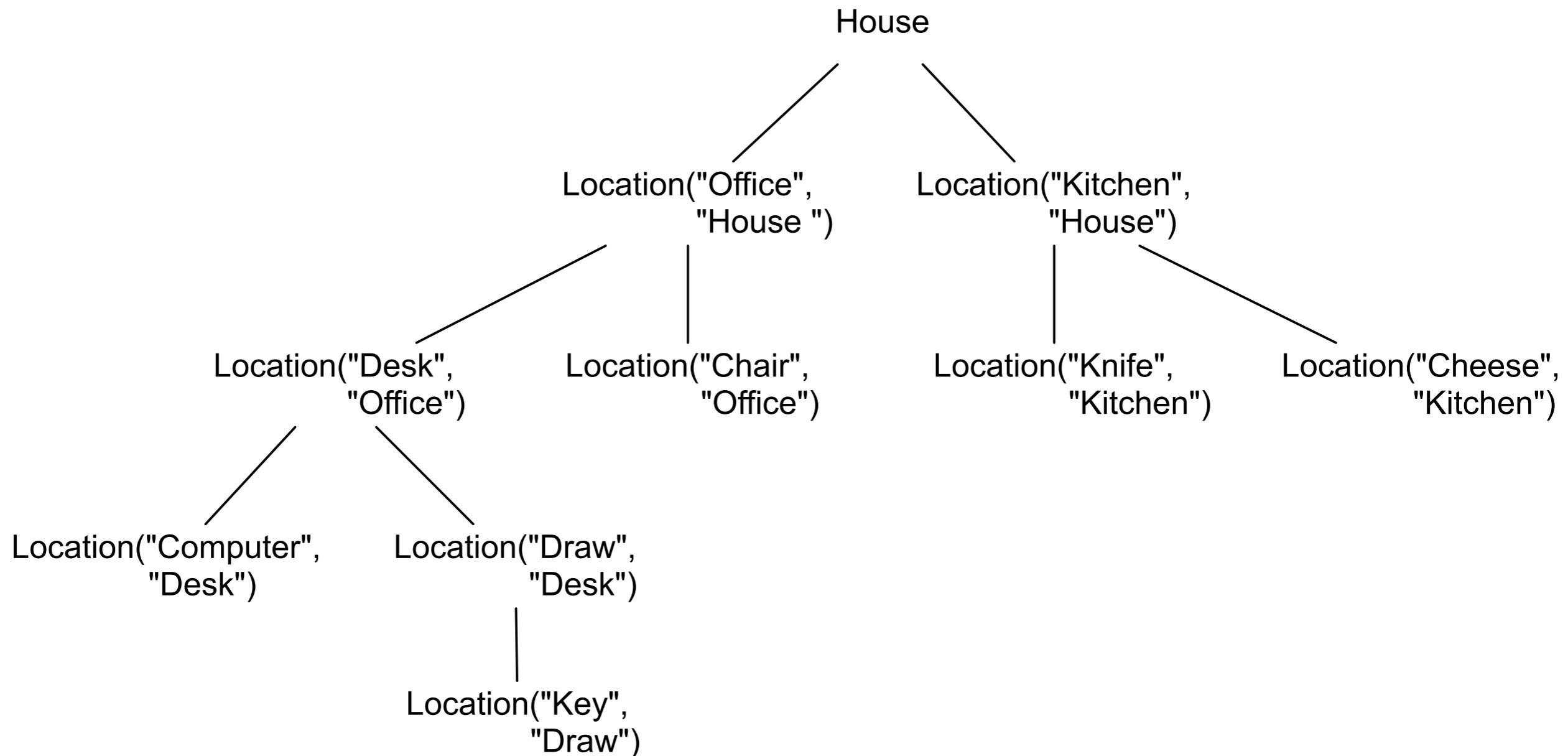
go4
thing Key is in the Office
thing Computer is in the Office
thing Draw is in the Office
thing Desk is in the Office
thing Chair is in the Office
```



Backward Chaining

```

rule "go5"
when
  String( this == "go5" )
  isContainedIn(thing, location; )
then
  System.out.println( "thing " + thing + " is in " + location );
end
  
```

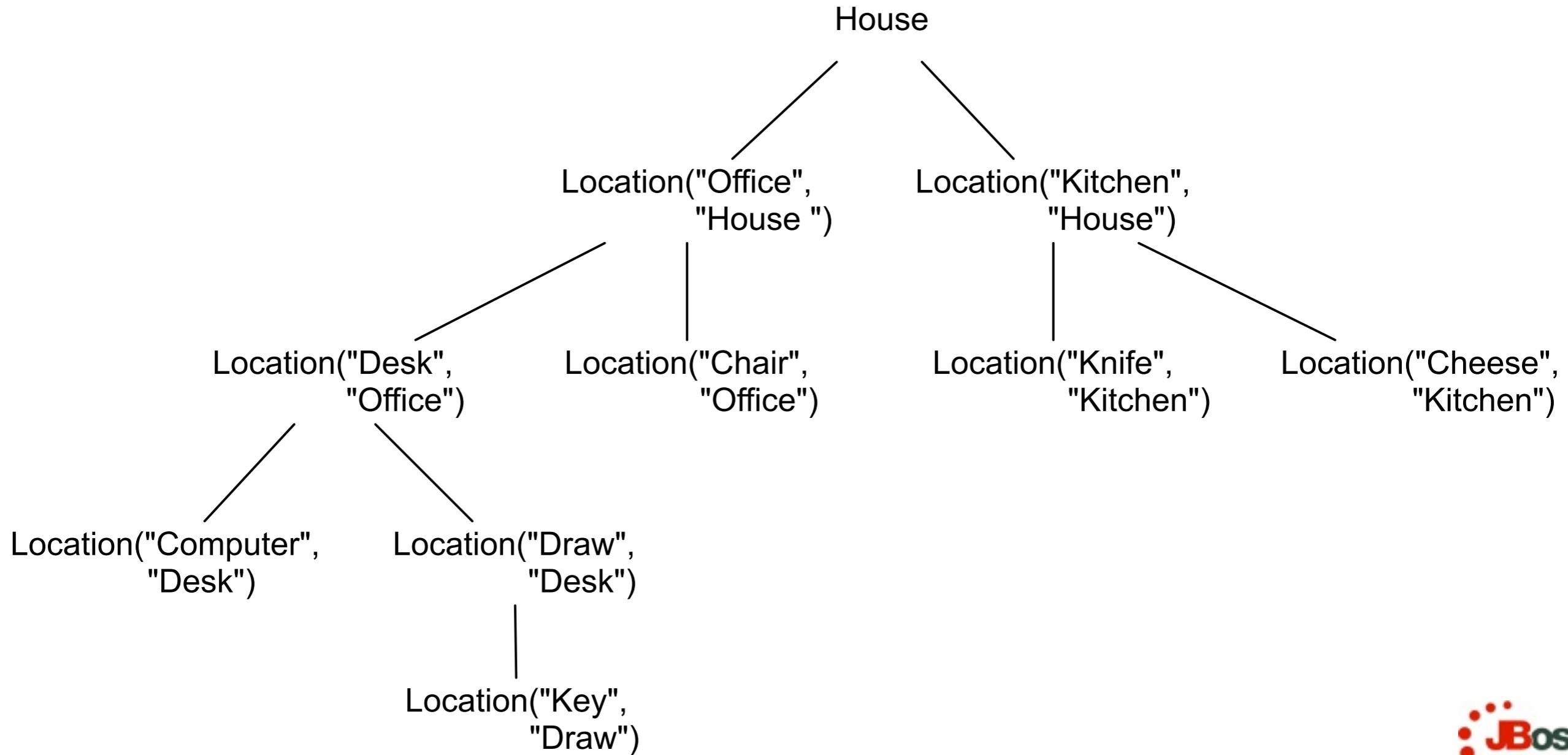


Backward Chaining

```

rule "go5"
when
  String( this == "go5" )
  isContainedIn(thing, location; )
then
  System.out.println( "thing " + thing + " is in " + location );
end

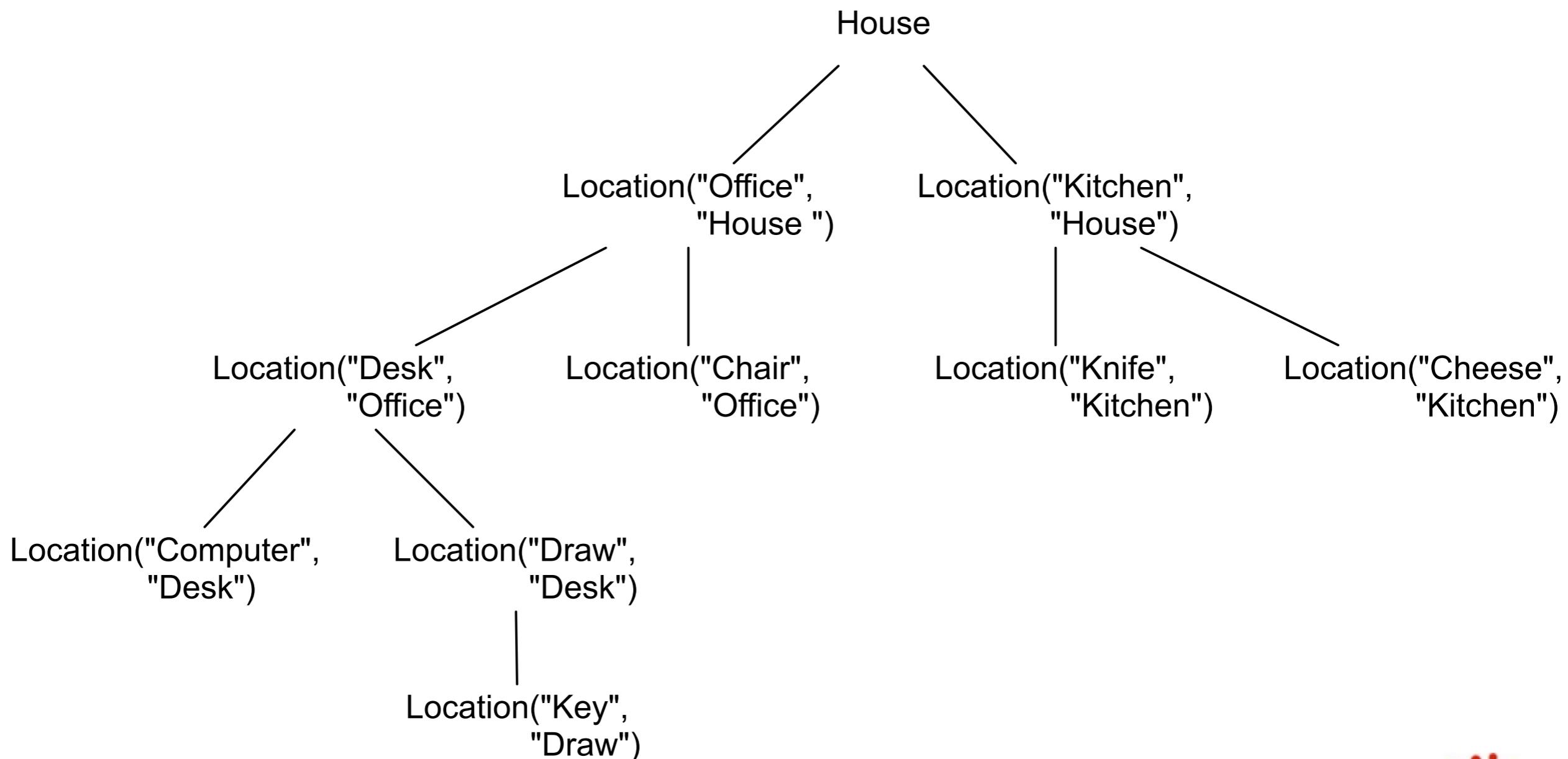
```



Backward Chaining

```

rule "go5"
when
  String( this == "go5" )
  isContainedIn(thing, location;)
then
  System.out.println( "thing " + thing + " is in " + location );
end
  
```



Backward Chaining

```

rule "go5"
when
  String( this == "go5" )
  isContainedIn(thing, location;)
then
  System.out.println( "thing " + thing + " is in " + location );
end
  
```

```

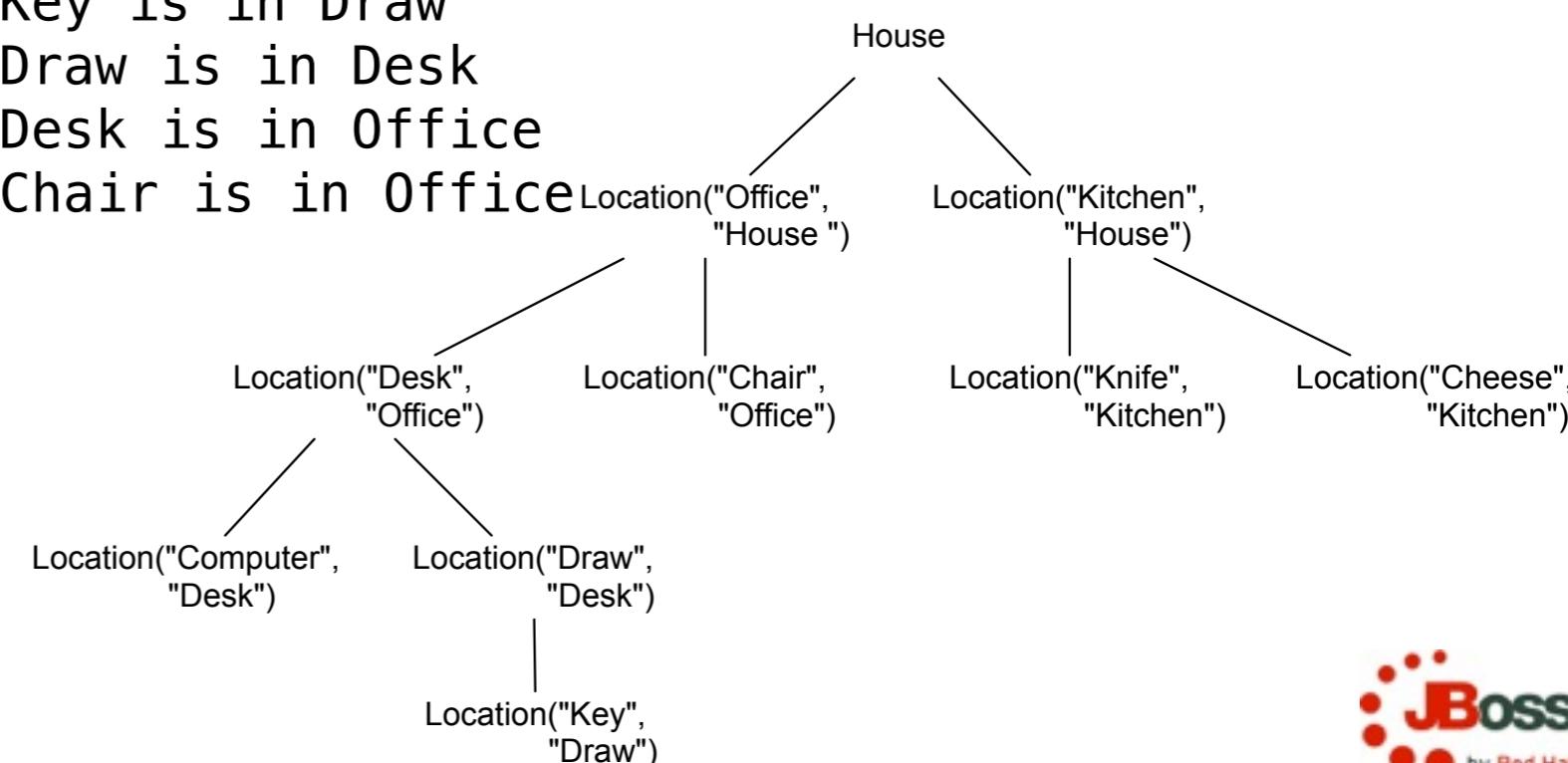
ksession.insert( "go5" );
ksession.fireAllRules();
---
```

```

go5
thing Knife is in House
thing Cheese is in House
thing Key is in House
thing Computer is in House
thing Draw is in House
thing Desk is in House
thing Chair is in House
thing Key is in Office
thing Computer is in Office
thing Draw is in Office
thing Key is in Desk
thing Office is in House
  
```

```

thing Computer is in Desk
thing Knife is in Kitchen
thing Cheese is in Kitchen
thing Kitchen is in House
thing Key is in Draw
thing Draw is in Desk
thing Desk is in Office
thing Chair is in Office
  
```



Questions?



- Dave Bowman: All right, HAL; I'll go in through the emergency airlock.
- HAL: Without your space helmet, Dave, you're going to find that rather difficult.
- Dave Bowman: HAL, I won't argue with you anymore! Open the doors!
- HAL: Dave, this conversation can serve no purpose anymore. Goodbye.

Joshya: Greetings, Professor Falken.

Falken: Hello, Joshua.

Joshya: A strange game. The only winning move is not to play. How about a nice game of chess?