

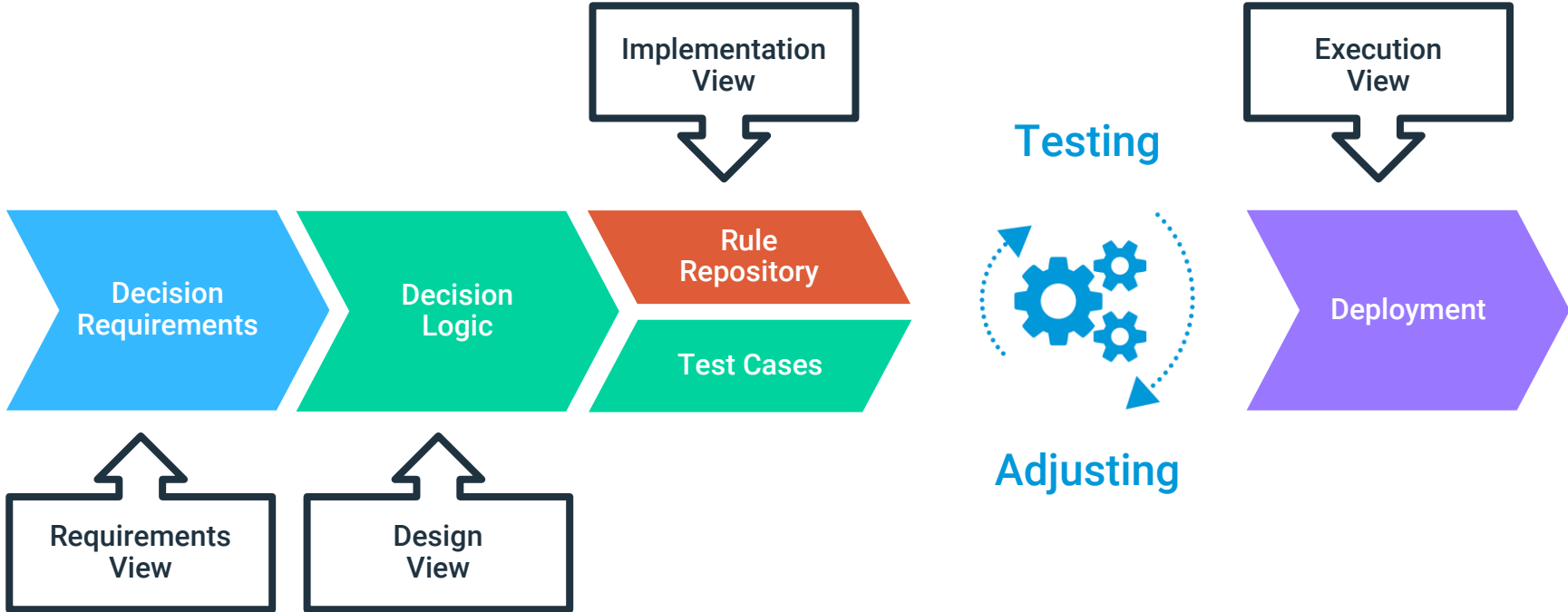
Creating Executable Decision Service Designs Using DMN and Excel Workbook

DecisionCAMP 2020

Gediminas Vedrickas

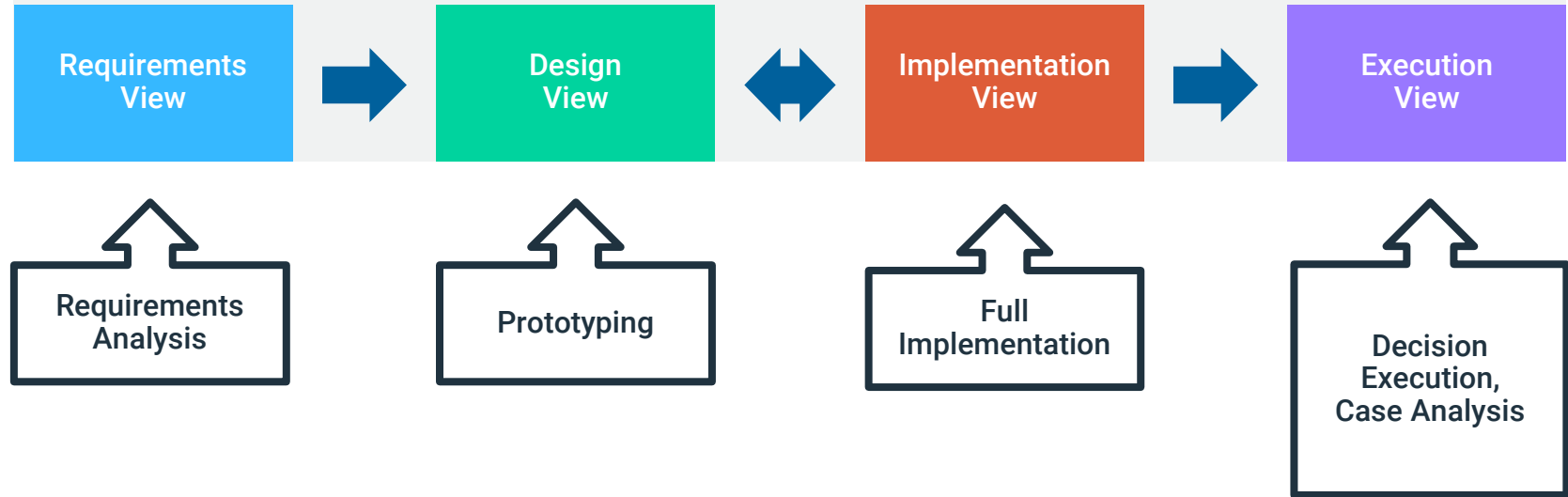
Director – Decision Solutions Practice Consulting,
FICO

Decision Implementation Lifecycle



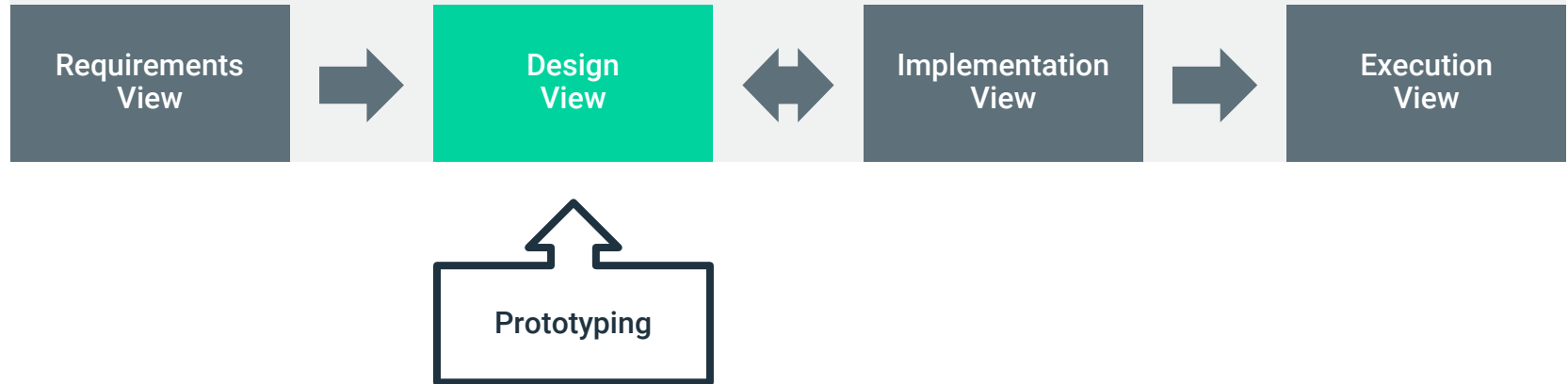
An Important Role for DMN

DMN as a shared Concept Model



Focus on Design View: Prototyping

DMN as a shared Concept Model



3 Steps of Prototyping a Decision Service

1

Performing Decision Analysis of Requirements Model

2

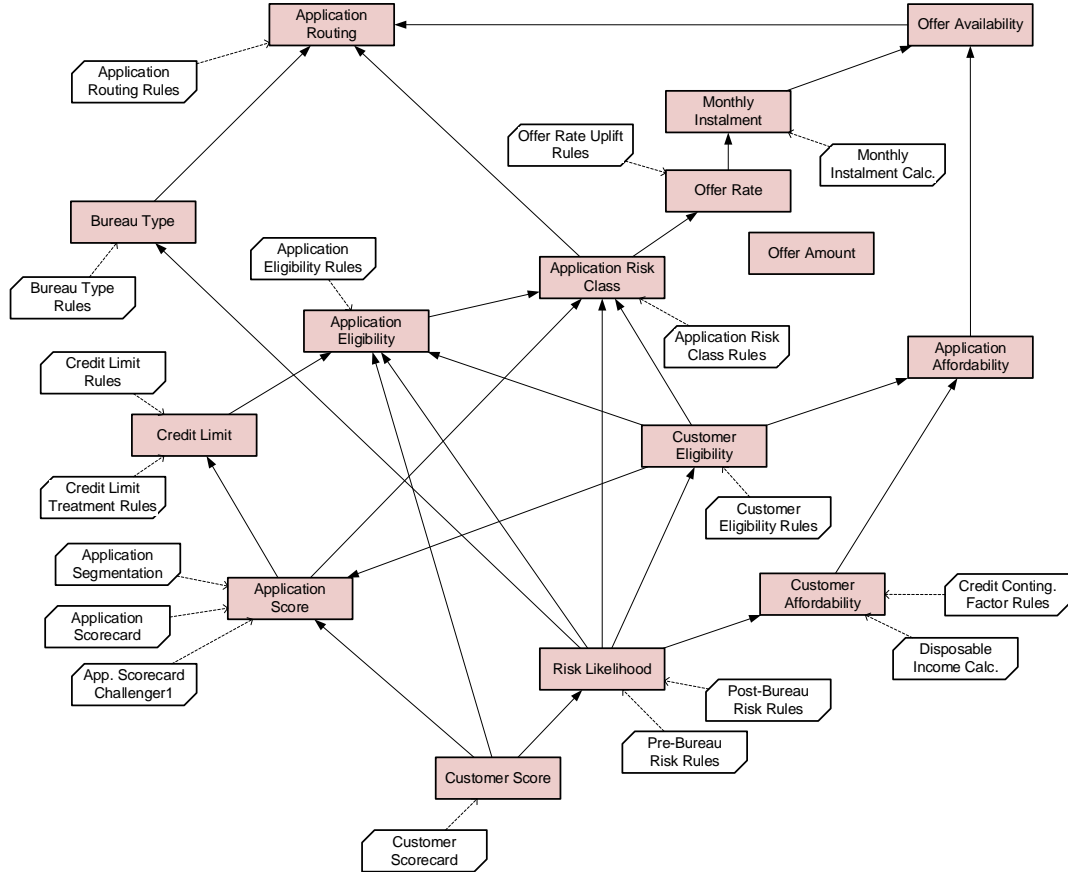
Creating/Refining Decision Model



3

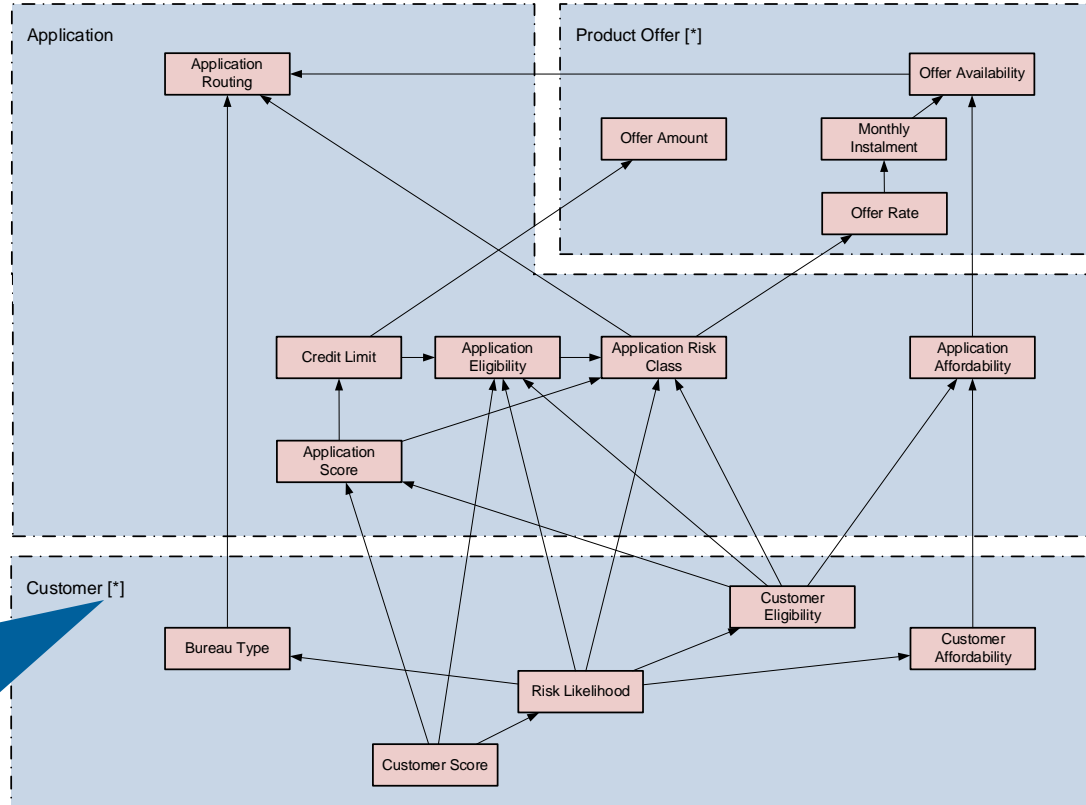
Capturing Decision Logic in Decision Workbook

Loan Decision Requirements



Step 1: Performing Decision Analysis

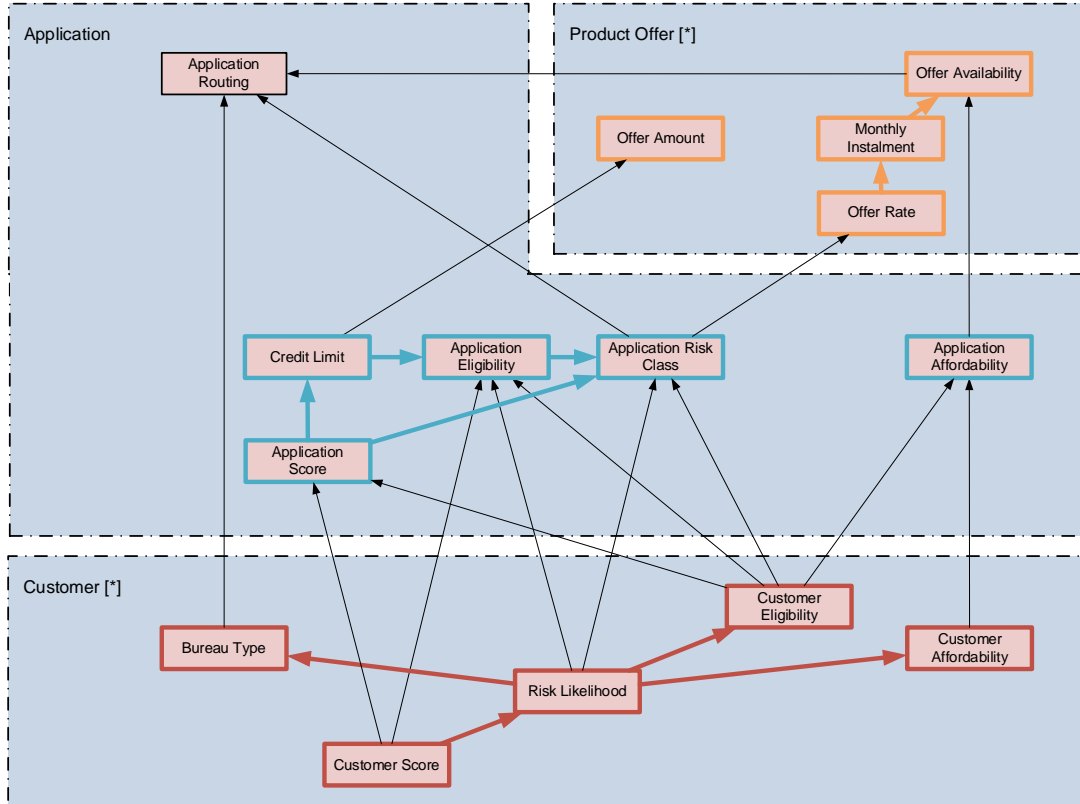
Grouping decisions by subject (decision level)



The specified multiplicity means *Customer* decisions will be made for each *Customer* individually

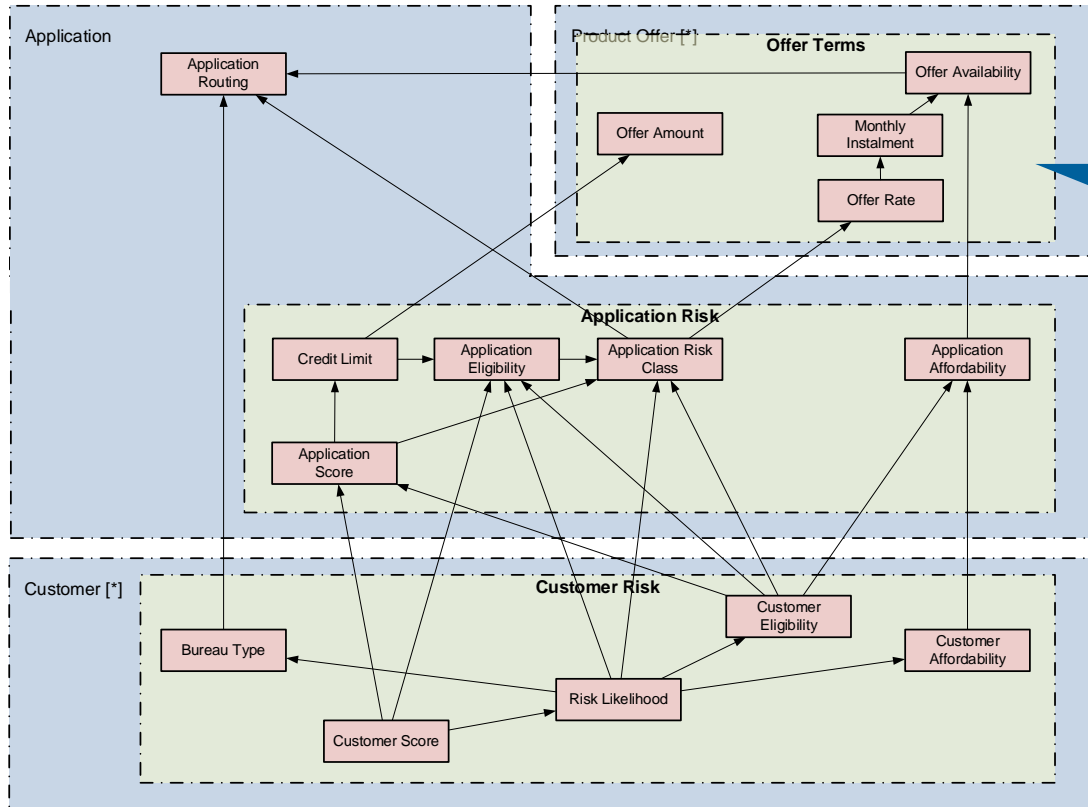
Step 1: Performing Decision Analysis

Grouping decisions by functional purpose



Step 1: Performing Decision Analysis

Grouping decisions by functional purpose

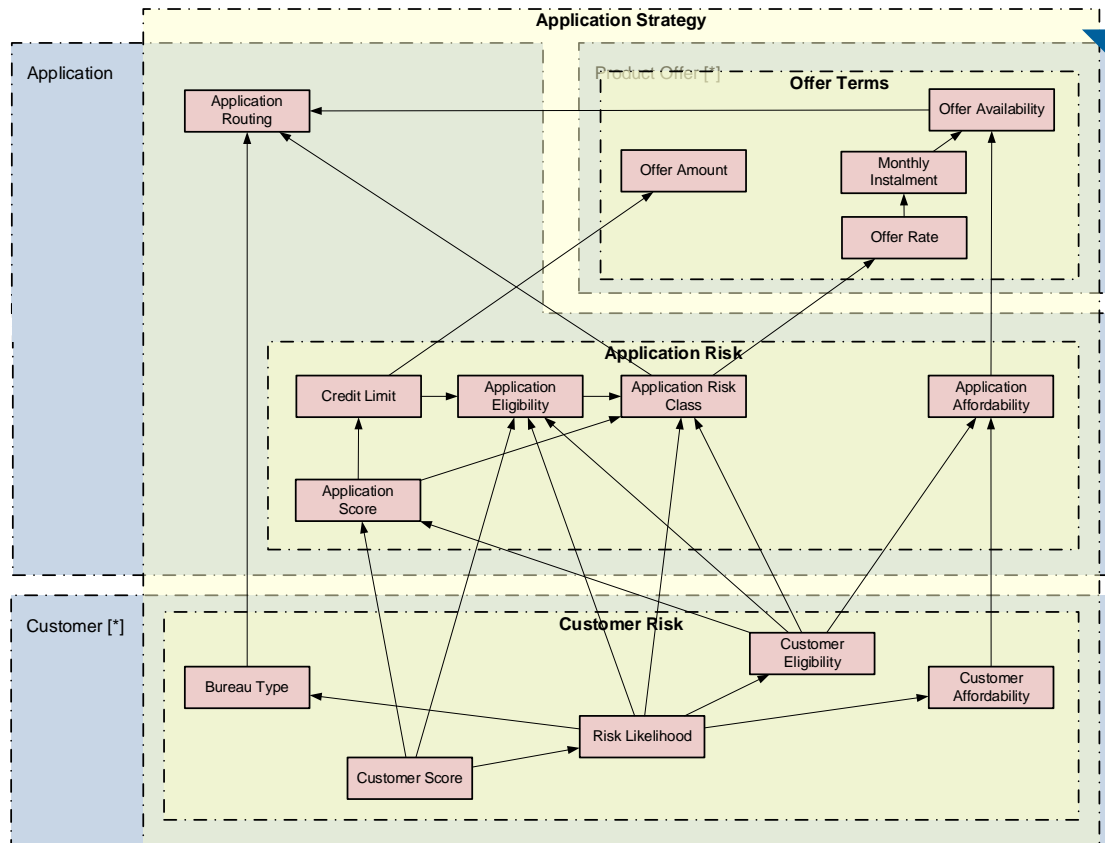


Offer terms related decisions form *Offer Terms* decision area

Decision Areas provide valuable higher-level view on how decision making is structured

Step 1: Performing Decision Analysis

Grouping decisions by functional purpose

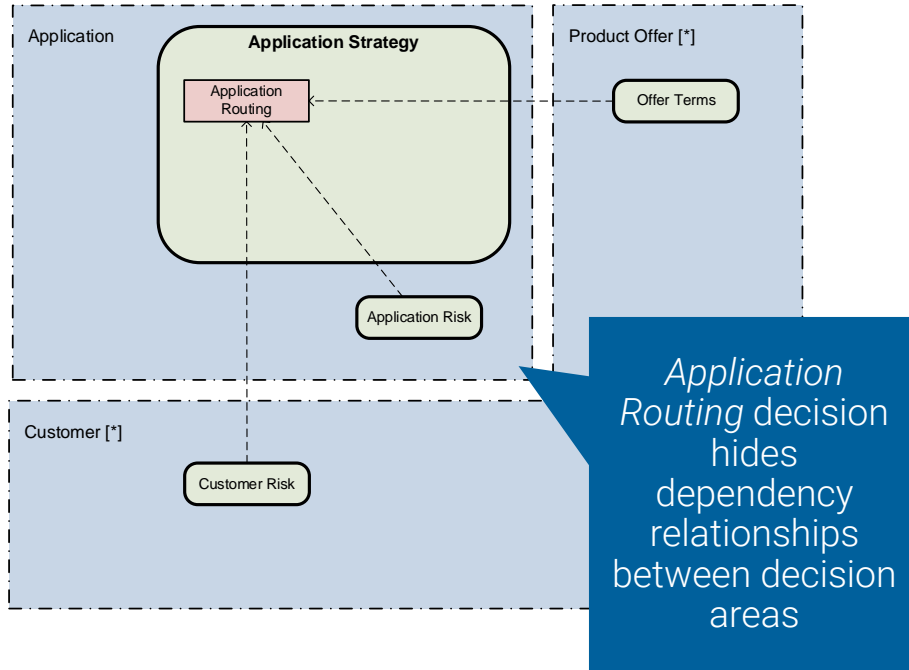


Decision Areas may not necessarily include decisions of the same subject, as it is the case with *Application Strategy*

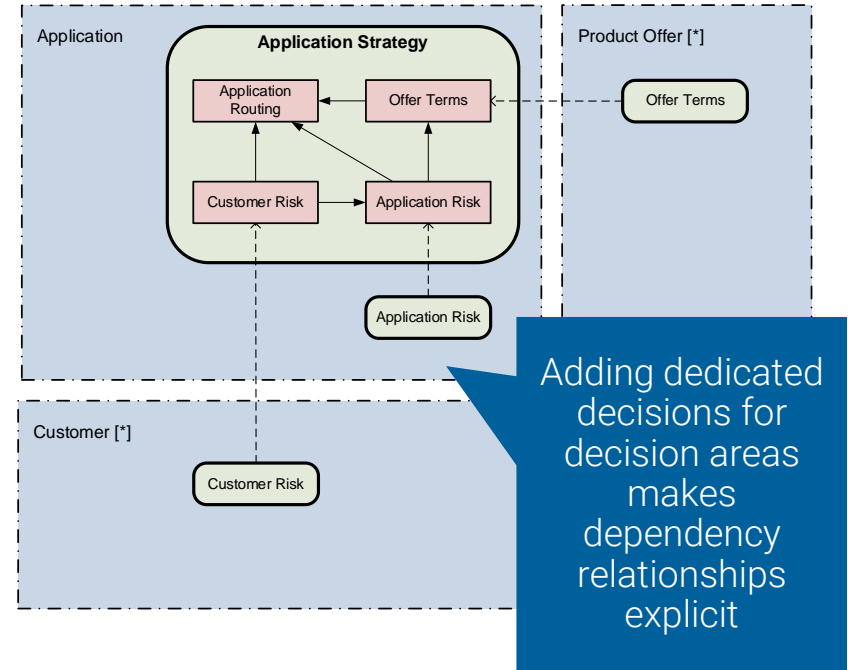
Step 2: Creating / Refining Decision Model

Options for implementing functional decision areas

Option 1



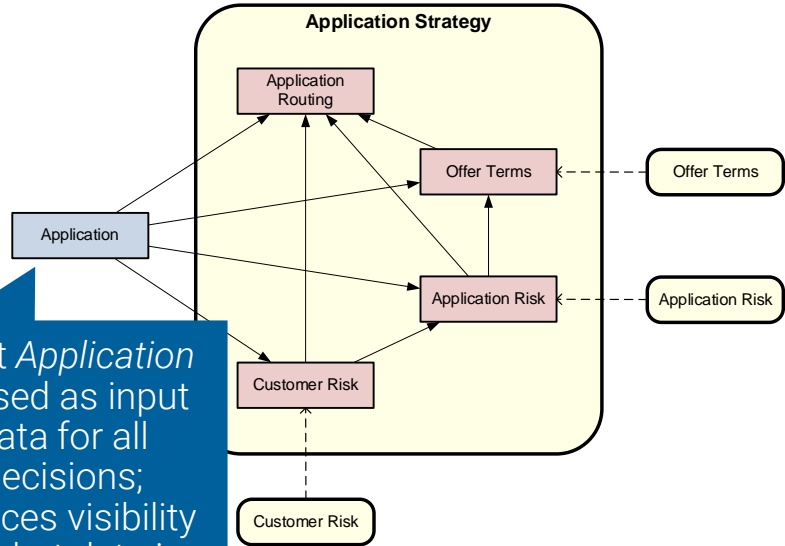
Option 2



Step 2: Creating / Refining Decision Model

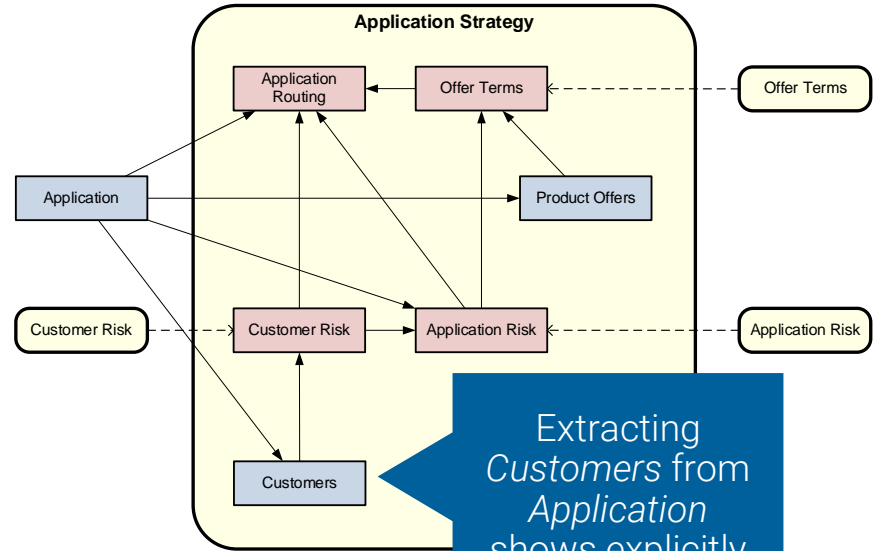
Options for implementing decision subjects (decision levels)

Option 1



Root *Application* is used as input data for all decisions; reduces visibility of what data is needed for a decision area

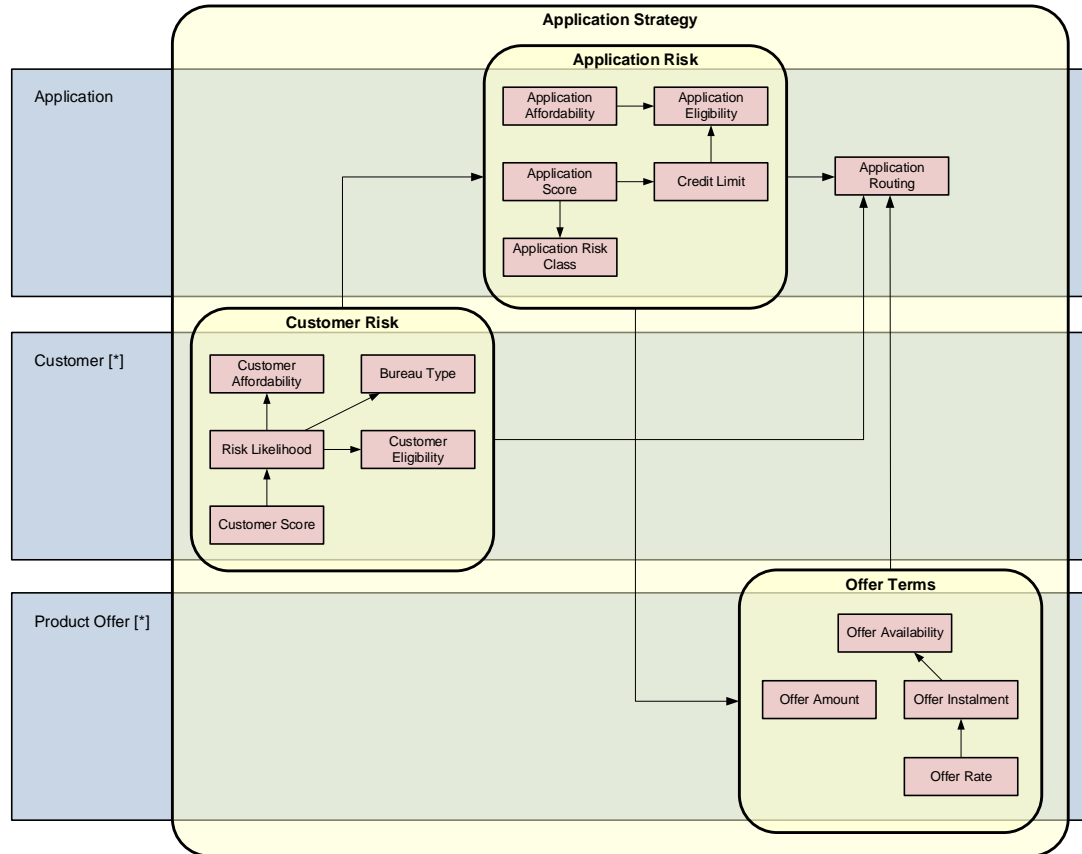
Option 2



Extracting *Customers* from *Application* shows explicitly what data is needed for the decision area

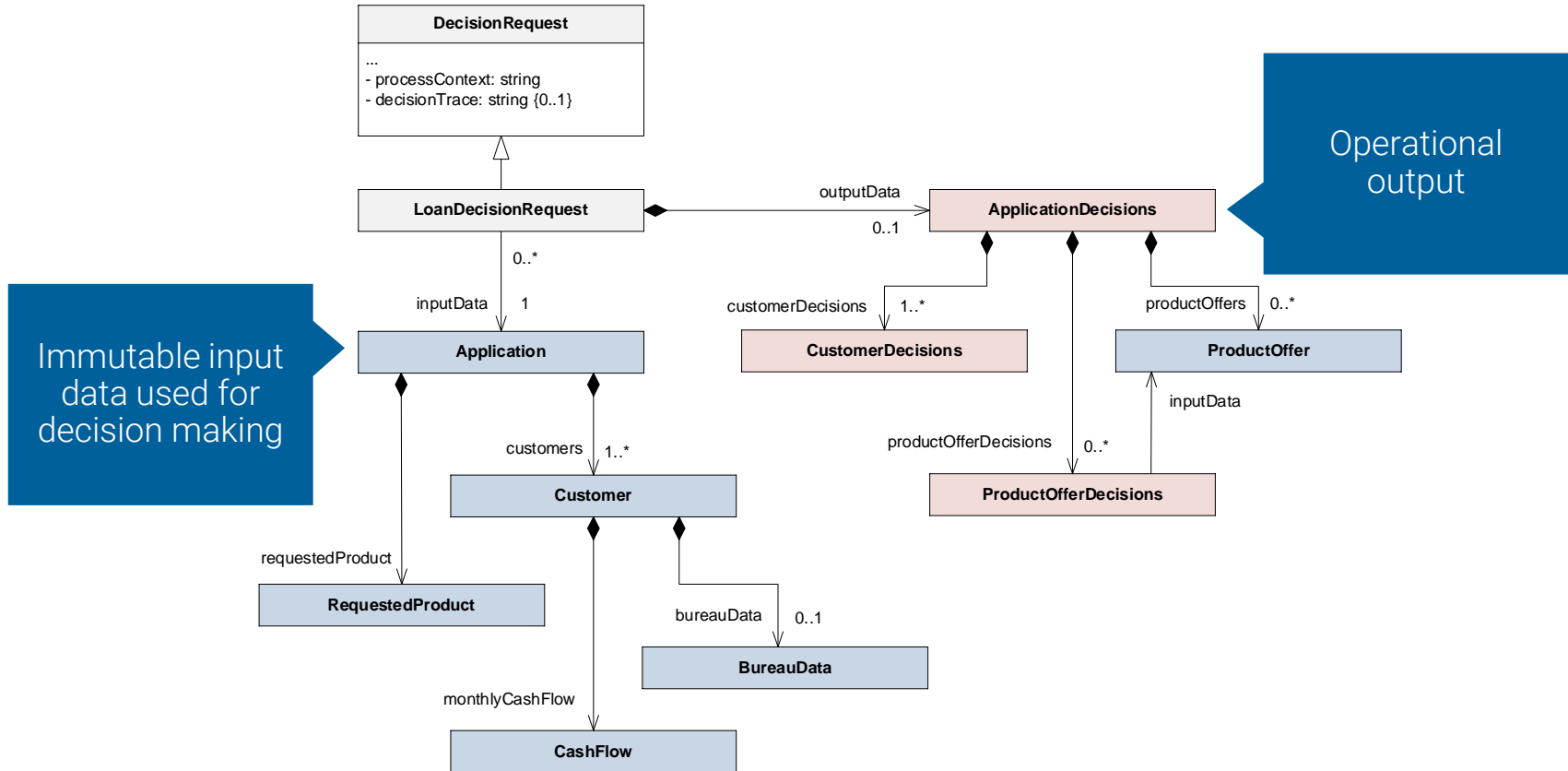
Step 2: Creating / Refining Decision Model

An informal view of the overall decision model



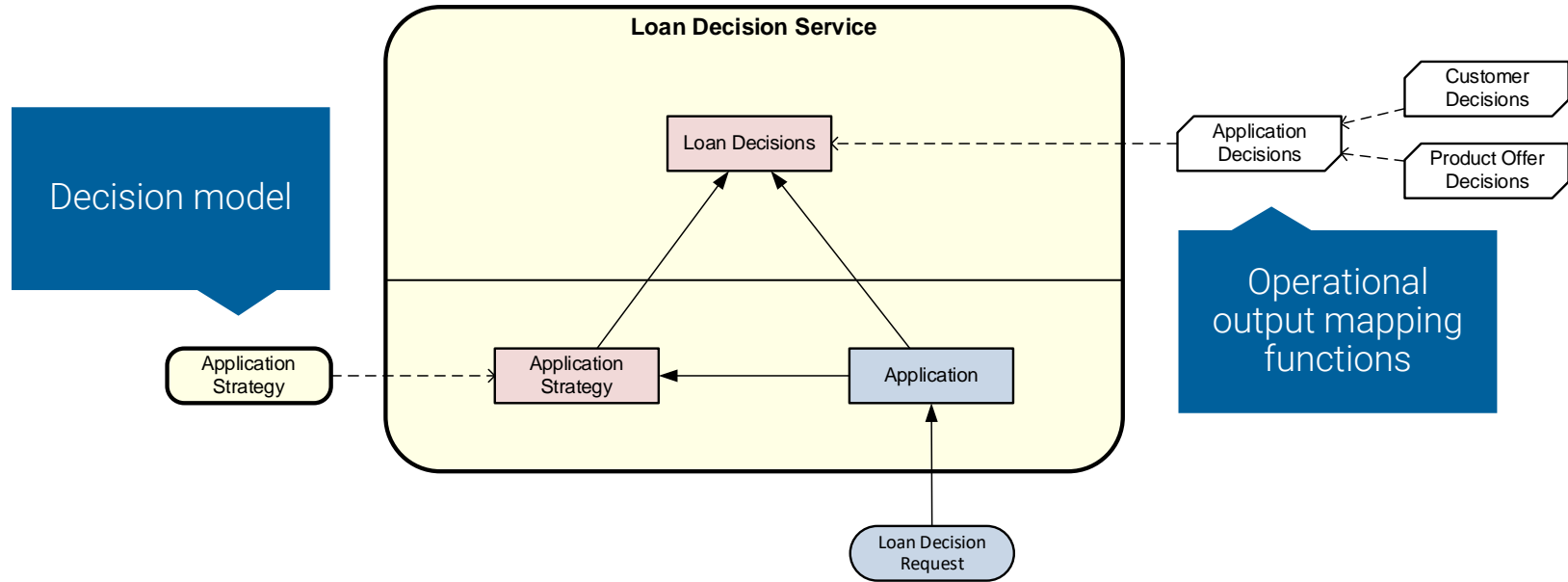
Step 3: Capturing Decision Logic in Decision Workbook

Decision Service Business Object Model (BOM)



Step 3: Capturing Decision Logic in Decision Workbook

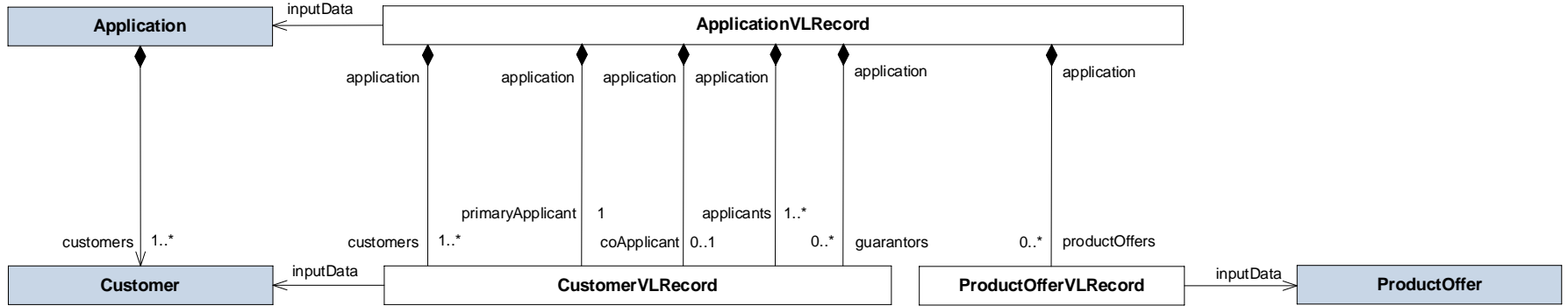
Loan Decision Service model



Step 3: Capturing Decision Logic in Decision Workbook

Variable Library

Variable Library records are used for extending immutable input data and enabling creating derived variables as well as projections over input data



Reference to the input data that was passed into decision service

Record may add bi-directional association relationships that are essential for simplifying decision logic expressions

Step 3: Capturing Decision Logic in Decision Workbook

General settings for global parameters

General settings for global parameters is a convenient technique for adjusting decision logic behavior at the service level

LoanPolicy
+ <i>static</i> loanTermOptions + <i>static</i> maxLoanTerm + <i>static</i> maxLoanAmount

CustomerPolicyTable
+ <i>static</i> reasonCode[*]

ApplicationPolicyTable
+ <i>static</i> reasonCode[*]

ContryCodeTable
+ <i>static</i> euCountryCode[*] + <i>static</i> countryCode[*]

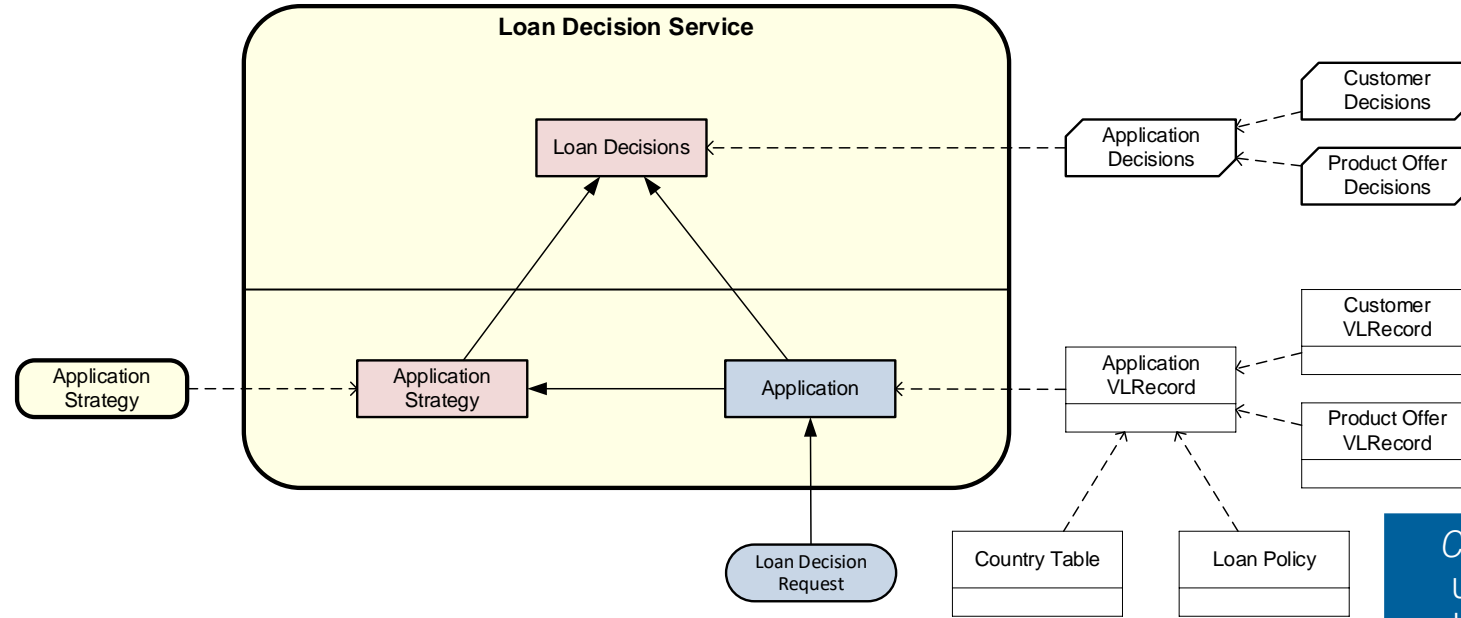
SpecialValueTable
+ <i>static</i> message[*]

For certain kind of parameters, e.g. general list of countries, may be unusual calling it a “decision”; though in DMN it is the only construct that allows to derive and to store value for use across decision model

Including global parameters into decision logic signature may unnecessarily complicate logic definition

Step 3: Capturing Decision Logic in Decision Workbook

Introducing *Class* construct, revising Loan Decision Service model



Class construct is used for both (a) extending immutable input data and (b) defining global parameters

Step 3: Capturing Decision Logic in Decision Workbook

Decision Workbook Demo

Loan Decision Service

Decision Service LoanDecisionService @ApplicationDecisions
Description goes here..
(loanDecisionRequest @LoanDecisionRequest)

application @ApplicationVlRecord
Description goes here..

Decision applicationStrategy @ApplicationStrategy
Description goes here..

ApplicationStrategy

application @ApplicationVlRecord application

Output Decisions

Decision loanDecisions @ApplicationDecisions
Description goes here..

Customer Risk Service

Decision Service CustomerRiskService @CustomerRisk
Description goes here..
F (customer @CustomerVlRecord)

Decision customerScore @real
Description goes here..

Decision customerRiskLikelihood @CustomerRiskLikelihood
Description goes here..

customerBehaviorRiskLikelihood @string **CustomerBehaviorRiskLikelihoodDecisionTable**(customer.inputData.existingCuston

bureauRiskLikelihood @string If customer.application.decisionPoint = "POB" //Post-Bureau
then **BureauRiskLikelihoodDecisionTable**(customerBehaviorRiskLikelihood, customer.input
else SpecialValueTable.message[message = "NotApplicable"].stringValue

riskLikelihood @string decisionPoint @string customer.application.decisionPoint

F	decisionPoint	riskLikelihood
1	PRB	customerBehaviorRiskLikelihood
2	POB	bureauRiskLikelihood
3	-	SpecialValueTable.message[message = "NotApplicable"].stringValue

//Post-Bureau

Decision customerEligibility @CustomerEligibility
Description goes here..

Decision customerAffordability @CustomerAffordability
Description goes here..

Customer Risk BKM

Decision Table CustomerBehaviorRiskLikelihoodDecisionTable @string
Description goes here..
F (isExistingCustomer @Integer, customerScore @real)

Decision Table BureauRiskLikelihoodDecisionTable @string
Description goes here..
F (customerBehaviorRiskLikelihood @string, creditScore @real)

U	customerBehaviorRiskLikelihood	customerScore	bureauRiskLikelihood @string
1	"High"	[0 .. 490]	"Very High"
2	"High"	[490 .. 690]	"High"
3	"High"	>= 690	"Medium"
4	"Medium"	[0 .. 490]	"High"
5	"Medium"	[490 .. 690]	"Medium"
6	"Medium"	>= 690	"Low"
7	"Low", "Very Low"	[0 .. 490]	"Medium"
8	"Low", "Very Low"	[490 .. 690]	"Low"
9	"Low", "Very Low"	>= 690	"Very Low"

Decision Table CreditContingencyFactorDecisionTable @real
Description goes here..
F (customerRiskLikelihood @CustomerRiskLikelihood)

Decision Table BureauCallTypeDecisionTable @string
Description goes here..
F (customerRiskLikelihood @CustomerRiskLikelihood, bureauReportAge @Integer, existingBureauCallType @string)

//Post-Bureau

Summary



- Prototyping is a valuable technique when creating decision services
- DMN and Excel Workbook provide formal background, tools and flexibility for creating executable decision service designs
- Using DMN as a shared concept model assures traceability of decision concepts and consistent user experience across decision implementation lifecycle
- DMN in its current maturity can address many real-world challenges, though some extensions would be valuable
 - Support for *Class* construct
 - Allowing *Range Variables* in invocation parameter list as a declarative alternative to iterating invocation over collection
 - New boxed expressions for *Rulesets*, *Scorecards*