ISDA CDM Use Cases
17 October 2019

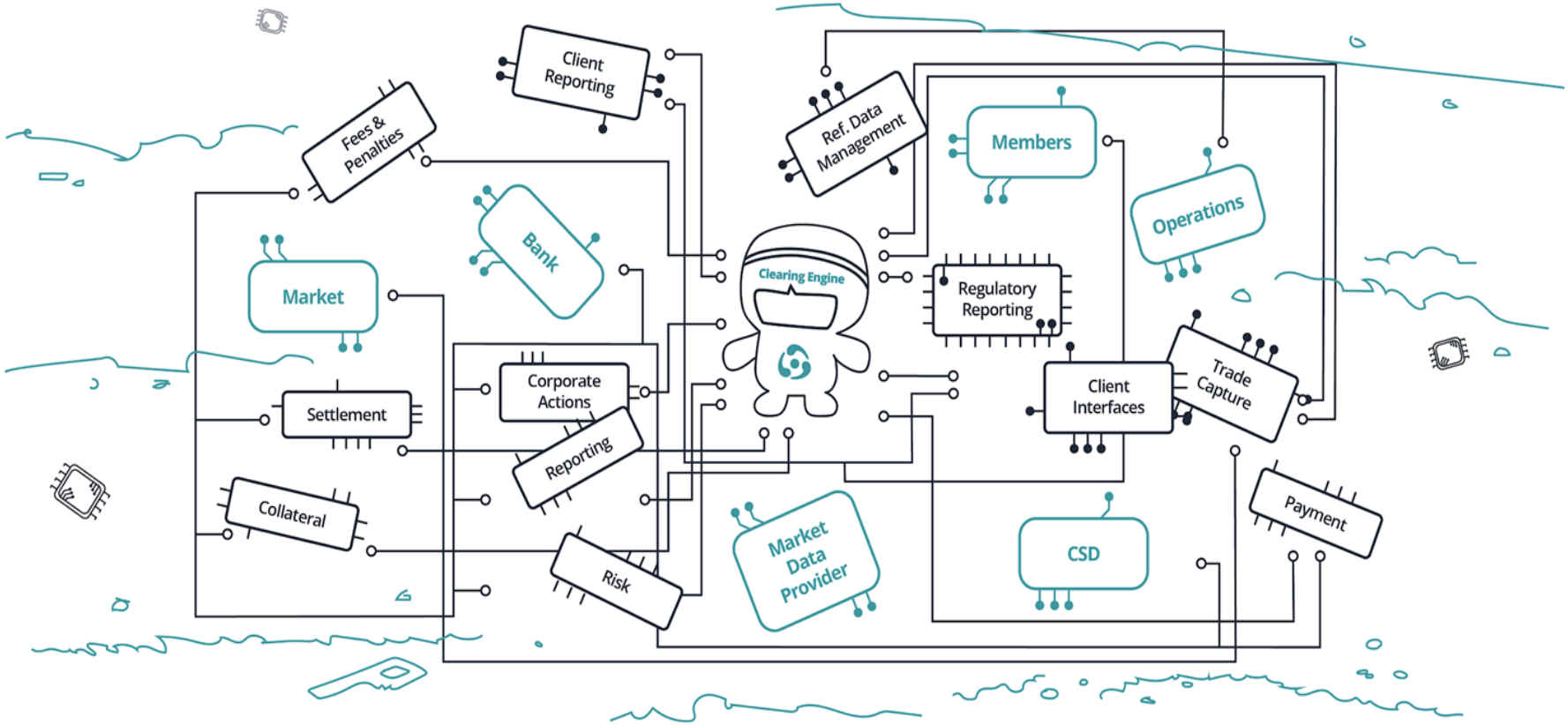
#DerivHack2019

Build Software to Test Software
exactpro.com

Contents

- Exactpro Team
- R3 Corda DLT – the Platform of Choice
- REGnosys/Rosetta our progress so far
- Implementing ISDA CDM on Corda
- Client Frontends
- Use Cases Walkthrough

Post Trade Technology Transformation Process



Award Winning Platform Software Testing Platform

(That we do not use for DerivHack)

exactpro
EXITUS ACTA PROBAT

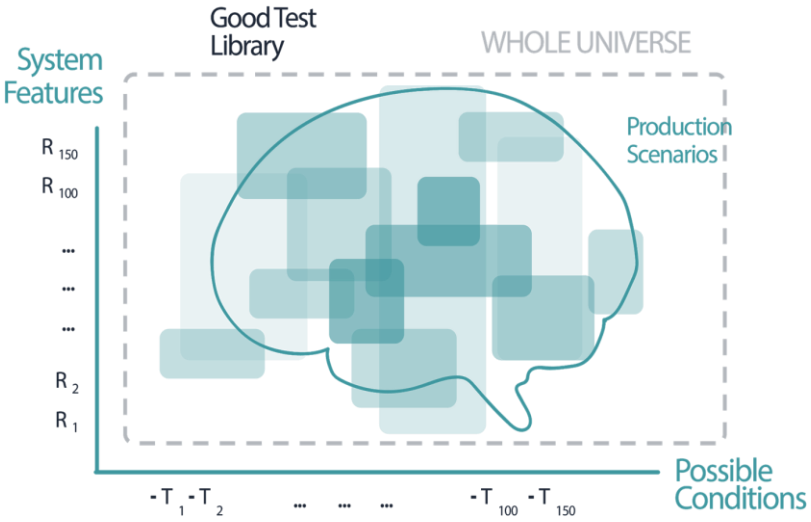
r3. corda

HYPERLEDGER FABRIC

DAML

IEEE Xplore® Digital Library

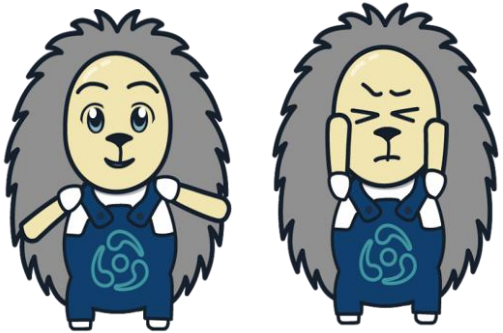
Build Software to Test Software



We learn from **Pain**, not from **Pleasure**

- Software Testing is Relentless Learning
- Automated Testing is Machine Learning
- Build the Next Generation of **Intelligent Software Testing Tools**

It is likely that there are several mechanisms responsible for intelligence in sapient beings. One of the them is semantic ontologies



Use ISDA CDM to Support Technology Transformation in Post Trade

We created a simple chat bot to promote this idea

DerivHack Takeaways on Technology Maturity

exactpro
EXITUS ACTA PROBAT

DLT Platform

r3. corda

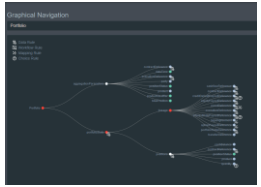
Perfect Choice

Domain Model

ISDA CDM

OK for Requirements Definition

Web Portal



There is still room for improvement

Code Generation

ROSETTA

NOT OK for Validations and Transitions

REGnosys/Rosetta Portal and Code Generation

Issue reported in Rosetta Portal: we've submitted a valid contribution for Allocate function, all checks were reported as passed: model compile, java generation, code compile.

Yet the system **failed to generate Allocate.java** (confirmed by Jim from REGnosys)

We've re-created the whole development workflow:

- built rosetta-dsl and rosetta-code-generators
- made changes in Rosetta portal
- downloaded Allocate.java and AffirmAllocatedTrade.java artifacts
- used it to produce a custom cdm-2.5.11_exactpro.jar
- built CordApp using obtained jar as a dependency
- developed version of AllocateImpl.java
- identified the issue in generated basis Allocate.java (**references are absent in Lineage object** – confirmed by Jim as a problem)
- worked around the issue above and implemented Use Cases

Rosetta Portal: Valid Syntax and All Checks are Passed – but Allocate.java is absent

exactpro
EXITUS ACTA PROBAT

No error message anywhere…

ROSETTA

IosifItkinTest | CDM

MODEL

- model-cdm-aliases
- model-cdm-calculations-csa
- model-cdm-calculations
- model-cdm-config
- model-cdm-event-confirmation
- model-cdm-event
- model-cdm-function-equity
- model-cdm-function-utilities
- **model-cdm-functions** M
- model-cdm-product
- model-derivatives-asset
- model-derivatives-bond-option
- model-derivatives-cd
- model-derivatives-doc
- model-derivatives-enum
- model-derivatives-eq-shared

```
alias allocationPrimitive :
  NewAllocationPrimitive( execution, allocationInstructions )

condition: <"Test. Check for security, copied from Settle function">
  execution -> product -> security exists

condition: <"Test. Breakdowns exist">
  allocationInstructions -> breakdowns exists

condition: <"Test. Error in code generation - no Allocation.java">
  allocationInstructions -> breakdowns count >= 0

assign-output allocationEvent -> primitive -> allocation:
  allocationPrimitive

  assign-output allocationEvent -> lineage -> eventReference:
      previousEvent

  assign-output allocationEvent -> lineage -> executionReference:
      execution

  assign-output allocationEvent -> eventEffect -> effectedExecution :
    allocationEvent -> primitive -> allocation -> after -> allocatedTrade -> execution
```

Contributed Workspace on Rosetta Technology Portal

exactpro
EXITUS ACTA PROBAT

ROSETTA

Press Me – and download Java code

losifItkinTest | CDM

MODEL

- model-cdm-aliases
- model-cdm-calculations-csa
- model-cdm-calculations
- model-cdm-config
- model-cdm-event-confirmation
- model-cdm-event
- model-cdm-function-equity
- model-cdm-function-utilities
- **model-cdm-functions** M
- model-cdm-product
- model-derivatives-asset
- model-derivatives-bond-option
- model-derivatives-cd
- model-derivatives-doc
- model-derivatives-enum
- model-derivatives-eq-shared

```
378  func AffirmAllocatedTrade : <"TBA Specify description">
379  inputs:
380    allocationEvent Event (1..1) <"Previous event for lineage purposes.">
381    allocatedTrade Trade (1..1) <"Trade to be affirmed.">
382    affirmationStatus AffirmationStatusEnum (1..1)
383
384  output:
385    affirmation Affirmation (1..1)
386
387  assign-output affirmation -> lineage -> eventReference :
388    allocationEvent
389
390  assign-output affirmation -> lineage -> executionReference :
391    allocatedTrade -> execution
392
393  assign-output affirmation -> status :
394    affirmationStatus
395
396  func ConfirmAllocatedTrade : <"TBA Specify description">
397  inputs:
398    allocationEvent Event (1..1) <"Previous event for lineage purposes.">
399    allocatedTrade Trade (1..1) <"Trade to be affirmed.">
400    confirmationStatus ConfirmationStatusEnum (1..1)
401
402  output:
```

Compare REGnosys GitHub vs. downloaded Java files

exactpro

{arclite:zip:losifltkinTest-java (1).zip:\src\generated\java\org\isda\cdm\functions} - Far 3.0.5151 x86

C:\Barclays\cdm-sources\org\isda\cdm\functions

n	Name	Name
..		ExtractContractState.java
example		ExtractQuantity.java
Abs.java		ExtractQuantityImpl.java
AbsImpl.java		FixedAmount.java
Allocate.java		FloatingAmount.java
AllocateImpl.java		FormContract.java
CalculationPeriod.java		FxMarkToMarket.java
CalculationPeriodImpl.java		GetRateSchedule.java
CdmToStrataMapper.java		GetRateScheduleImpl.java
DayCountFraction.java		GreaterThan.java
Equals.java		GreaterThanEquals.java
EqualsImpl.java		GreaterThanEqualsImpl.java
EquityAmountPayer.java		GreaterThanImpl.java
EquityAmountPayerImpl.java		InterpolateForwardRate.java
EquityCalculationPeriodImpl.java		InterpolateForwardRateImpl.java
EquityCashSettlementAmount.java		NewAllocationPrimitive.java
EquityNotionalAmount.java		NewAllocationPrimitiveImpl.java
EquityPerformance.java		NewCashTransferPrimitive.java
EquityPriceObservation.java		NewContractEventImpl.java
EquityPriceObservationImpl.java		NewContractFormationFromExecution}
EquitySpot.java		NewContractFormationPrimitive.jav}
EquitySpotImpl.java		NewEquitySwapProduct.java
EvaluatePortfolioState.java		NewExecutionFromProductImpl.java
EvaluatePortfolioStateImpl.java		NewExecutionPrimitive.java
Execute.java		NewFloatingPayout.java

.. Up 01/10/19 16:52

Bytes: 185267, files: 88, folders: 1

...-java (1).zip:\src\generated\java\org\isda\cdm\functions 10:34

n	Name	Name
..		NewAllocationPrimitive.java
Abs.java		NewCashTransferPrimitive.java
AffirmAllocatedTrade.java		NewContractFormationPrimitive.jav
Allocate.java		NewEquitySwapProduct.java
CalculationPeriod.java		NewExecutionPrimitive.java
DayCountFraction.java		NewFloatingPayout.java
Equals.java		NewQuantityChangePrimitive.java
EquityAmountPayer.java		NewResetPrimitive.java
EquityCashSettlementAmount.java		NewSingleNameEquityPayout.java
EquityNotionalAmount.java		NewTransferPrimitive.java
EquityPerformance.java		PeriodsInYear.java
EquityPriceObservation.java		Plus.java
EquitySpot.java		QuantityEquals.java
EvaluatePortfolioState.java		RateOfReturn.java
Execute.java		Reset.java
ExtractContractState.java		ResolveAdjustableDate.java
ExtractQuantity.java		ResolveCashflow.java
FixedAmount.java		ResolveCashSettlementDate.java
FloatingAmount.java		ResolveEquityContract.java
FormContract.java		ResolveEquityPeriodEndPrice.java
FxMarkToMarket.java		ResolveEquityPeriodStartPrice.jav
GetRateSchedule.java		ResolvePrice.java
GreaterThan.java		ResolveQuantity.java
GreaterThanEquals.java		ResolveRateIndex.java
InterpolateForwardRate.java		ResolveTimeZoneFromTimeType.java

.. Up 14/10/19 12:22

Bytes: 172125, files: 55, folders: 0

\src\generated\java\org\isda\cdm\functions>

Compare REGnosys GitHub vs. downloaded Java files

exactpro
EXITUS ACTA PROBAT

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack [C:\dev\derivhack] - ...\dev\derivhack\lib\dependencies\cdm-2.5.11_exactpro.jar]

derivhack 〉 lib 〉 dependencies 〉 cdm-2.5.11_exactpro.jar 〉 org 〉 isda 〉 cdm 〉 functions 〉 Allocate

Add Configuration... Git:

Project

- TransferFlow.kt
- resources
- test
- build.gradle
- doc
- gradle
- lib
  - dependencies [test] test resources root
    - aopalliance-1.0.jar
    - cdm-2.5.11_exactpro.jar
      - available-samples
      - calculation-test-cases
      - cdm
      - cdm-sample-files
      - com
      - config
      - distribution
      - distribution-daml
      - META-INF
      - org.isda.cdm
        - blueprint
        - builders
        - functions
          - example.services.identification
          - Abs
          - AbsImpl
          - AffirmAllocatedTrade

va | SettleImpl.java | NewTransferPrimitiveImpl.java | Allocate.class | AffirmAllocatedTrade.class | AllocateImpl.class | Execu

Decompiled .class file, bytecode version: 52.0 (Java 8) Download... Choose Sources...

```
32  @   public Allocate() {
33      }
34
35      public Event evaluate(Execution execution, AllocationInstructions allocationInstructions, Event previousEvent) {
36          assert ValidatorHelper.exists(MapperS.of(execution).map( name: "getProduct", Execution::getProduct).map( name: "gets
37
38          assert ValidatorHelper.exists(MapperS.of(allocationInstructions).mapC( name: "getBreakdowns", AllocationInstructions
39
40          EventBuilder allocationEventHolder = this.doEvaluate(execution, allocationInstructions, previousEvent);
41          Event allocationEvent = this.assignOutput(allocationEventHolder, execution, allocationInstructions, previousEvent).bu
42          this.objectValidator.validateAndFailOnErorr(Event.class, allocationEvent);
43          return allocationEvent;
44      }
45
46  @   private EventBuilder assignOutput(EventBuilder allocationEventHolder, Execution execution, AllocationInstructions allocat
47          Event allocationEvent = allocationEventHolder.build();
48          allocationEventHolder.getOrCreatePrimitive().addAllocation((AllocationPrimitive)MapperS.of(this.allocationPrimitive(
49          allocationEvent = allocationEventHolder.build();
50          allocationEventHolder.getOrCreateLineage().addEventReferenceRef((Event)MapperS.of(previousEvent).get());
51          allocationEvent = allocationEventHolder.build();
52          allocationEventHolder.getOrCreateLineage().addExecutionReferenceRef((Execution)MapperS.of(execution).get());
53          allocationEvent = allocationEventHolder.build();
54          allocationEventHolder.getOrCreateEventEffect().addEffectedExecutionRef(MapperS.of(allocationEvent).map( name: "getPri
55          return allocationEventHolder;
56      }
```

Pre-Event R3 Platform Instructions Code

exactpro
EXITUS ACTA PROBAT

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack-corda-orig - ...\CdmValidators.kt [derivhack-corda-orig.cdm-support.main]

hack-corda-orig 〉 cdm-support 〉 src 〉 main 〉 kotlin 〉 net 〉 corda 〉 cdmsupport 〉 validators 〉 CdmValidators.kt

Add Configuration... Git:

Project

- derivhack-corda-orig C:\dev\derivhack-corda-or
  - .gradle
  - .idea
  - allSampleFiles
  - cdm-support
    - src
      - main
        - java
          - implementations
            - CordaRuntimeModule
            - PortfolioInstructions
        - kotlin
          - net.corda.cdmsupport
            - builders
            - eventparsing
            - extensions
            - functions
            - transactionbuilding
            - validators
              - CdmValidators
            - vaultquerying
              - CdmVaultQuery.kt
            - Exceptions.kt
        - resources
      - test
    - build.gradle

ExecutionState.kt | ConfirmationBuilderFromExecution.kt | CdmVaultQuery.kt | CdmValidators.kt | ConfirmationStatusEnum.class

```kotlin
class CdmValidators() {

    fun validateExecution(execution: Execution): List<ValidationResult<in Execution>> {
        val executionMeta = ExecutionMeta()
        val validators = ArrayList<Validator<in Execution>>()
        validators.addAll(executionMeta.choiceRuleValidators())
        validators.addAll(executionMeta.dataRules())
        validators.add(executionMeta.validator())

        return validators.map { it.validate(RosettaPath.valueOf( stringPath: "Execution"), execution) }.toList()
    }

    fun validateEvent(event: Event): List<ValidationResult<in Event>> {
        val eventMeta = EventMeta()
        val validators = ArrayList<Validator<in Event>>()
        validators.addAll(eventMeta.choiceRuleValidators())
        validators.addAll(eventMeta.dataRules())
        validators.add(eventMeta.validator())

        return validators.map { it.validate(RosettaPath.valueOf( stringPath: "Event"), event) }.toList()
    }

    fun validateExecutionPrimitive(executionPrimitive: ExecutionPrimitive): List<ValidationResult<in ExecutionPrimitive>> {
        val executionPrimitiveMeta = ExecutionPrimitiveMeta()
        val validators = ArrayList<Validator<in ExecutionPrimitive>>()
        validators.addAll(executionPrimitiveMeta.choiceRuleValidators())
```

Our Re-Factoring

exactpro

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack [C:\dev\derivhack] - ...\validators\CdmValidators.kt [derivhack.cdm-support.main]

derivhack 〉 cdm-support 〉 src 〉 main 〉 kotlin 〉 net 〉 corda 〉 cdmsupport 〉 validators 〉 CdmValidators.kt Add Configuration... Git:

Project

- derivhack C:\dev\derivhack
  - .gradle
  - .idea
  - allSampleFiles
  - cdm-support
    - src
      - main
        - java
          - implementations
            - AffirmAllocatedTradeImpl
            - AllocateImpl
            - ConfirmAllocatedTradeImpl
            - CordaRuntimeModule
            - EvaluatePortfolioStateCordaImpl
            - NewAllocationPrimitiveImpl
            - NewTransferPrimitiveImpl
            - SettleImpl
          - PortfolioInstructions
        - kotlin
        - resources
      - test

CdmValidators.kt

```kotlin
3   import ...
11
12  class CdmValidators() {
13
14      fun validateEvent(event: Event) {
15          getValidators(EventMeta()).map { it.validate(RosettaPath.valueOf( stringPath: "Event"), event) }.forEach { it: ValidationRe
16              if (!it.isSuccess) {
17                  throw RuntimeException(it.failureReason.get())
18              }
19          }
20      }
21
22      fun <T : RosettaModelObject> getValidators(meta: RosettaMetaData<in T>): List<Validator<in T>> {
23          val validators = ArrayList<Validator<in T>>()
24          validators.addAll(meta.choiceRuleValidators())
25          validators.addAll(meta.dataRules())
26          validators.add(meta.validator())
27          return validators
28      }
```

Version Control: Local Changes Log Console Update Info: 16/10/2019 11:01

Branch: All User: All Date: All Paths: All

Fix upload allocation instruction REST	origin & dev	dzavodchikov	16/10/2019 10:55
Add confirmation flow test, fix states and contracts		julia	16/10/2019 10:39
Fix upload allocation instruction REST		dzavodchikov	16/10/2019 10:38

Select commit to view details

Implement Validations not yet supported in

exactpro
EXITUS ACTA PROBAT

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack [C:\dev\derivhack] - ...\validators\CdmValidators.kt [derivhack.cdm-support.main]

derivhack 〉 cdm-support 〉 src 〉 main 〉 kotlin 〉 net 〉 corda 〉 cdmsupport 〉 validators 〉 CdmValidators.kt Add Configuration... Git:

Project

- derivhack C:\dev\derivhack
  - .gradle
  - .idea
  - allSampleFiles
  - cdm-support
    - src
      - main
        - java
          - implementations
            - AffirmAllocatedTradeImpl
            - AllocateImpl
            - ConfirmAllocatedTradeImpl
            - CordaRuntimeModule
            - EvaluatePortfolioStateCordaImpl
            - NewAllocationPrimitiveImpl
            - NewTransferPrimitiveImpl
            - SettleImpl
          - PortfolioInstructions
        - kotlin
        - resources
      - test
    - build.gradle
  - cdm-support-contracts-states
    - src
      - main
        - kotlin

CdmValidators.kt AllocateImpl.java

```
80          }
81          validateExecution(executionPrimitive.after.execution)
7507    /*********************
7508     * Event Qualification
7509     */
7510
7511    isEvent Allocation <"The qualification of allocation event from the fact that (i) the in
            (iv) there is a single contractualProduct before (ensured through cardinality). Note
7512        Event -> primitive -> allocation only exists
7513        /*
7514         * TODO - The below syntax needs to be expanded to specify a summation
7515         * This could be specify as quantityBeforeAllocation = sum(quantityAfterAllocation)
7516         */
7517    //  and quantityBeforeAllocation = quantityAfterAllocation
7518
95              throw RuntimeException("Wrong identifier for allocated trade")
96          }
97      }
98
99      val totalAllocatedQty = allocationPrimitive.after.allocatedTrade.map { it.execution.quantity.amount }.reduce(BigDec
100     if (originalTrade.execution.quantity.amount != totalAllocatedQty) {
101         throw RuntimeException("Wrong total allocated qty")
102     }
103
104     allocationPrimitive.after.allocatedTrade.forEach { it: Trade!
105         if (it.execution.partyRole.single { it.role == PartyRoleEnum.CLIENT } == originalTrade.execution.partyRole.singl
```

Pre-Event R3 Platform Instructions Code

- The CDM defines a number of key classes:
  - `Execution` A class to specify an execution, which consists essentially in the economic terms which are agreed between the parties, alongside with the qualification of the type of execution.
  - `Event` A class to specify the lifecycle event.
  - `TransferPrimitive` A class to specify the transfer of assets between parties, those assets being either cash, securities or physical assets.
  - `AllocationPrimitive` A class to specify the primitive event to represent a split/allocation of a trade.
  - `Affirmation` A class to specify a trade affirmation.
  - `Confirmation` A class to specify a trade confirmation.

Pre-Event R3 Platform Instructions

exactpro
EXITUS ACTA PROBAT

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack [C:\dev\derivhack] - ...\kotlin\com\derivhack\TransferFlow.kt [derivhack.cordapp.main]

derivhack 〉 cordapp 〉 src 〉 main 〉 kotlin 〉 com 〉 derivhack 〉 TransferFlow.kt

Add Configuration... Git:

Project

- helpers
- models
- Controller
- NodeRPCConnection.kt
- Server.kt
- Client.kt
- build.gradle
- config
- consolerpc
- cordapp
  - src
    - main
      - kotlin
        - com.derivhack
          - AffirmationFlow.kt
          - AllocationFlow.kt
          - ConfirmationFlow.kt
          - ExecutionFlow.kt
          - ObserveryFlow.kt
          - PortfolioFlow
          - PrintVaultFlow
          - TransferFlow.kt
      - resources
    - test
  - build.gradle
- doc

ExecutionState.kt | CdmValidators.kt | EventState.kt | TransferFlow.kt | AllocationFlow.kt | ConfirmationFlow.kt

```kotlin
38
39      progressTracker.currentStep = CDMFlowTracker.Companion.RUN_VALIDATORS
40      val cdmValidator = CdmValidators()
41      cdmValidator.validateEvent(transferEvent)
42      transferEvent.primitive.transfer.forEach { cdmValidator.validateTransferPrimitive(it) }
43
44      val notary = serviceHub.networkMapCache.notaryIdentities.first()
45
46      progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION
47      val cdmTransactionBuilder = CdmTransactionBuilder(notary, transferEvent, DefaultCdmVaultQuery(serviceHub), progressT
48
49      progressTracker.currentStep = CDMFlowTracker.Companion.VERIFY
50      cdmTransactionBuilder.verify(serviceHub)
51
52      progressTracker.currentStep = CDMFlowTracker.Companion.SIGN_INITIAL_TRANSACTION
53      val signedByMe = serviceHub.signInitialTransaction(cdmTransactionBuilder)
54
55      val regulator = serviceHub.identityService.partiesFromName( query: "Observery",  exactMatch: true).single()
56
57      progressTracker.currentStep = CDMFlowTracker.Companion.GET_SIGNING_PARTIES
58      val counterPartySessions = cdmTransactionBuilder.getPartiesToSign().minus(ourIdentity)
59              .map { initiateFlow(it) }
60
61      progressTracker.currentStep = CDMFlowTracker.Companion.COLLECTING_SIGNATURES
62      val fullySignedTx = subFlow(CollectSignaturesFlow(signedByMe, counterPartySessions, CollectSignaturesFlow.tracker()))
63
```

Corda Flows Implementation – up to UC 5

exactpro
EXITUS ACTA PROBAT

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help derivhack [C:\dev\derivhack] - ...\kotlin\com\derivhack\TransferFlow.kt [derivhack.cordapp.main]

derivhack 〉 cordapp 〉 src 〉 main 〉 kotlin 〉 com 〉 derivhack 〉 TransferFlow.kt

Add Configuration... Git:

Project

- helpers
- models
- Controller
- NodeRPCConnection.kt
- Server.kt
- Client.kt
- build.gradle
- config
- consolerpc
- cordapp
  - src
    - main
      - kotlin
        - com.derivhack
          - AffirmationFlow.kt
          - AllocationFlow.kt
          - ConfirmationFlow.kt
          - ExecutionFlow.kt
          - ObserveryFlow.kt
          - PortfolioFlow
          - PrintVaultFlow
          - TransferFlow.kt
      - resources
    - test
  - build.gradle
- doc

ExecutionState.kt | CdmValidators.kt | EventState.kt | TransferFlow.kt | AllocationFlow.kt | ConfirmationFlow.kt

```kotlin
38
39      progressTracker.currentStep = CDMFlowTracker.Companion.RUN_VALIDATORS
40      val cdmValidator = CdmValidators()
41      cdmValidator.validateEvent(transferEvent)
42      transferEvent.primitive.transfer.forEach { cdmValidator.validateTransferPrimitive(it) }
43
44      val notary = serviceHub.networkMapCache.notaryIdentities.first()
45
46      progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION
47      val cdmTransactionBuilder = CdmTransactionBuilder(notary, transferEvent, DefaultCdmVaultQuery(serviceHub), progressT
48
49      progressTracker.currentStep = CDMFlowTracker.Companion.VERIFY
50      cdmTransactionBuilder.verify(serviceHub)
51
52      progressTracker.currentStep = CDMFlowTracker.Companion.SIGN_INITIAL_TRANSACTION
53      val signedByMe = serviceHub.signInitialTransaction(cdmTransactionBuilder)
54
55      val regulator = serviceHub.identityService.partiesFromName( query: "Observery",  exactMatch: true).single()
56
57      progressTracker.currentStep = CDMFlowTracker.Companion.GET_SIGNING_PARTIES
58      val counterPartySessions = cdmTransactionBuilder.getPartiesToSign().minus(ourIdentity)
59              .map { initiateFlow(it) }
60
61      progressTracker.currentStep = CDMFlowTracker.Companion.COLLECTING_SIGNATURES
62      val fullySignedTx = subFlow(CollectSignaturesFlow(signedByMe, counterPartySessions, CollectSignaturesFlow.tracker()))
63
```

Technicality – use Allocation command instead of Execution

exactpro
EXITUS ACTA PROBAT

Project

- helpers
- models
- Controller
- NodeRPCConnection.kt
- Server.kt
- Client.kt
- build.gradle
- config
- consolerpc
- cordapp
  - src
    - main
      - kotlin
        - com.derivhack
          - AffirmationFlow.kt
          - AllocationFlow.kt
          - ConfirmationFlow.kt
          - ExecutionFlow.kt
          - ObserveryFlow.kt
          - PortfolioFlow
          - PrintVaultFlow
          - TransferFlow.kt
      - resources
    - test
  - build.gradle
- doc

CdmTransactionBuilder.kt

```kotlin
124 private fun processAllocationPrimitive(allocationPrimitive: AllocationPrimitive) {
139         val outputOriginalState = createExecutionState(allocationPrimitive.after.originalTrade.execution)
140
141         progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION_4
142         addOutputState(outputOriginalState, CDMEvent.ID)
143
144         addOutputState(EventState(serializeCdmObjectIntoJson(event), inputState.state.data.participants))
145
146         progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION_5
147         var keys: Set<PublicKey> = outputOriginalState.participants.map { p -> p.owningKey }.toSet()
148
149         progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION_6
150         val outputAllocatedStates = allocationPrimitive.after.allocatedTrade.map { createExecutionStateFromAfterAllocatio
151
152         progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION_7
153         outputAllocatedStates.forEach { it: ExecutionState
154             addOutputStateReturnIndex(it.copy(workflowStatus = "ALLOCATED"), CDMEvent.ID)
155             keys = keys.plus(it.participants.map { p -> p.owningKey })
156         }
157
158         progressTracker.currentStep = CDMFlowTracker.Companion.BUILD_TRANSACTION_8
159         addCommand(CDMEvent.Commands.Allocation(), keys.toList())
160     }
161 }
162
163 private fun processTransferPrimitive(transferPrimitive: TransferPrimitive) {
164     val outputTransferState = createTransferState(transferPrimitive)
```

R3 Corda: It looks like a deliberately planted defect in R3 basis version offered to participants who use Corda

exactpro
EXITUS ACTA PROBAT

States are not stored into Observery vault…

```
20  @InitiatedBy(TestObservableFlow::class)
21  class ReceiveObservableFlow(private val otherSideSession: FlowSession) : FlowLogic<Unit>() {
22      @Suspendable
23      override fun call() {
24          // Start the matching side of SendTransactionFlow above, but tell it to record all visible states even
25          // though they (as far as the node can tell) are nothing to do with us.
26          subFlow(ReceiveTransactionFlow(otherSideSession, checkSufficientSignatures: false, StatesToRecord.ALL_VISIBLE))
```

ExecutionState.kt | CdmVaultQuery.kt | TestObservableFlow.kt | ReceiveTransactionFlow.kt

```
20  *
21  * Please note that it will *not* store the transaction to the vault unless that is explicitly requested and checkSufficientSignatures is true.
22  * Setting statesToRecord to anything else when checkSufficientSignatures is false will *not* update the vault.
23  *
24  * @property otherSideSession session to the other side which is calling [SendTransactionFlow].
25  * @property checkSufficientSignatures if true checks all required signatures are present. See [SignedTransaction.verify].
26  * @property statesToRecord which transaction states should be recorded in the vault, if any.
27  */
28  open class ReceiveTransactionFlow @JvmOverloads constructor(private val otherSideSession: FlowSession,
29                                                              private val checkSufficientSignatures: Boolean = true,
30                                                              private val statesToRecord: StatesToRecord = StatesToRecord.NONE) : FlowLogic<SignedTransact
```

R3 Corda: CDM Orthodoxy

Commit Changes

Changelist: Default Changelist

Git

derivhack.cdm-support.main 1 file C:\dev\derivhack\cdm-support\src\main

kotlin\net\corda\cdmsupport\vaultquerying 1 file

CdmVaultQuery.kt

dev 1 modified

Author:

Amend commit

Sign-off commit

Commit Message

Before Commit

Reformat code

Rearrange code

Optimize imports

CDM Orthodoxy - we should use WorkflowStatusEnum as it is referred from Event instead of AffirmationStatus or ConfirmationStatus

Diff

Side-by-side viewer | Do not ignore | Highlight words

2 differenc

07da3eb7130d0e1ff18524fb190b956bfe8e8f5b

Your version

```
urn serviceHub.vaultService
    .queryBy<ExecutionState>().states
    .filter { it.state.data.workflowStatus == AffirmationStatusEnum.AFFIRMED.name }
    .map { ex -> ex.state.data.execution() }

le fun getConfirmedExecutions(): List<Execution> {

urn serviceHub.vaultService
    .queryBy<ExecutionState>().states
    .filter { it.state.data.workflowStatus == ConfirmationStatusEnum.CONFIRMED.name
    .map { ex -> ex.state.data.execution() }

le fun getAffirmations(): List<Affirmation> {
```

```
41  41  n serviceHub.vaultService
42  42      .queryBy<ExecutionState>().states
43  43      .filter { it.state.data.workflowStatus == WorkflowStatusEnum.AFFIRMED.name }
44  44      .map { ex -> ex.state.data.execution() }
45  45
46  46
47  47  fun getConfirmedExecutions(): List<Execution> {
48  48
49  49  n serviceHub.vaultService
50  50      .queryBy<ExecutionState>().states
51  51      .filter { it.state.data.workflowStatus == WorkflowStatusEnum.CONFIRMED.name }
52  52      .map { ex -> ex.state.data.execution() }
53  53
54  54
55  55  fun getAffirmations(): List<Affirmation> {
```

Commit | Cancel

Clients to connect to our Corda Network

exactpro
EXITUS ACTA PROBAT

r3test.tlp - r3test@10.44.59.12:22 - Bitvise xterm - r3test@r3-exposed-srv09: ~/derivhack

```
---------------- ClientDetails(nodeAddress=10.44.59.19:10052, rpcUsername=client2, rpcPassword=test) ----------------
Starting
Parse Event
Run Validators
Transactional builder
Transactional builder 1
Transactional builder 2
Transactional builder 3
Transactional builder 4
Transactional builder 5
Transactional builder 6
Transactional builder 7
Verify transaction
Sign initial transaction
Get signing parties
Collecting signatures from other parties
Collecting signatures from counterparties.
Starting
Verifying collected signatures.
Run Finality Flow
Requesting signature by notary service
Requesting signature by Notary service
Validating response from Notary service
Broadcasting transaction to participants
Run Observery Flow
Starting
---------------- ClientDetails(nodeAddress=10.44.59.19:10002, rpcUsername=notary, rpcPassword=test) ----------------
Done
nExecutions=0 Consumed=0
nAffirmations=0 Consumed=0
---------------- ClientDetails(nodeAddress=10.44.59.19:10012, rpcUsername=client1, rpcPassword=test) ----------------
nExecutions=6 Consumed=4
nAffirmations=4 Consumed=0
```

Console Router

Chat bot

127.0.0.1:5000

Chatbot

Are you a bear?

Great, carry on!

bear?

I am not a bear

Type a message and

Barclays Derivhack - Exactpro

broker1

Execution

Affirmation

Confirmation

Transfer

Portfolio

Filter

Consumed Unconsumed

Upload

row	executionType	executionVenue	price_accruedInterest	quantity_amount	linearId
0	ELECTRONIC	Execution Venue	1.3700	800000.00	8f8fb522-9d11-48b2-9e6c-c53.
1	ELECTRONIC	Execution Venue	1.3700	800000.00	1f517b62-4f35-4877-a603-5d3.
2	ELECTRONIC	Execution Venue	1.3700	800000.00	9d13a93a-fb28-444b-af4f-c998
3	ELECTRONIC	Execution Venue	1.3700	800000.00	a6c0f89d-d271-4ce0-aef5-c237
4	ELECTRONIC	Execution Venue	1.3700	800000.00	57bc3fb9-0e7d-4275-b10c-07f.
5	ELECTRONIC	Execution Venue	1.3700	800000.00	7e9f5167-1b72-407e-b1c0-e11.
6	ELECTRONIC	Execution Venue	1.3700	800000.00	1b413e90-e4e2-48bd-b7c6-777
7	ELECTRONIC	Execution Venue	1.3700	800000.00	96d7b339-45c6-45cc-a79b-59a
8	ELECTRONIC	Execution Venue	1.0400	300000.00	5ca7c295-8209-47f3-af73-5323
9	ELECTRONIC	Execution Venue	1.3700	800000.00	97fb1ec8-aea4-4207-9999-25b.
10	ELECTRONIC	Execution Venue	1.3700	320000.00	00a5dcd3-0ae6-4e51-a39e-975
11	ELECTRONIC	Execution Venue	1.3700	480000.00	2287ac24-9b80-40a6-89c4-1ff0
12	ELECTRONIC	Execution Venue	1.3700	320000.00	3bf6d813-e1e8-4507-8127-28e
13	ELECTRONIC	Execution Venue	1.3700	480000.00	df521ba3-c6f5-4dbf-8aa6-8625

Vault Level Web Front-End

r3test.tlp - r3test@10.44.59.12:22 - Bitvise xterm - r3test@r3-exposed-srv09: ~/derivhack

```
---------------- ClientDetails(nodeAddress=10.44.59.19:10052, rpcUsername=client2, rpcPassword=test) ----------------
Starting
Parse Event
Run Validators
Transactional builder
Transactional builder 1
Transactional builder 2
Transactional builder 3
Transactional builder 4
Transactional builder 5
Transactional builder 6
Transactional builder 7
Verify transaction
Sign initial transaction
Get signing parties
Collecting signatures from other parties
Collecting signatures from counterparties.
Starting
Verifying collected signatures.
Run Finality Flow
Requesting signature by notary service
Requesting signature by Notary service
Validating response from Notary service
Broadcasting transaction to participants
Run Observery Flow
Starting
---------------- ClientDetails(nodeAddress=10.44.59.19:10002, rpcUsername=notary, rpcPassword=test) ----------------
Done
nExecutions=0 Consumed=0
nAffirmations=0 Consumed=0
---------------- ClientDetails(nodeAddress=10.44.59.19:10012, rpcUsername=client1, rpcPassword=test) ----------------
nExecutions=6 Consumed=4
nAffirmations=4 Consumed=0
```

```kotlin
fun getExecutionFlowProgressTracker() : ProgressTracker {
    return ProgressTracker(
            PARSE_EVENT, RUN_VALIDATORS, BUILD_TRANSACTION,
            VERIFY, SIGN_INITIAL_TRANSACTION, GET_SIGNING_PARTIES,
            COLLECTING_SIGNATURES, RUN_FINALITY_FLOW, RUN_OBSERVERY_FLOW)
}
```

Vault Level Front End

exactpro
EXITUS ACTA PROBAT

Barclays Derivhack - Exactpro

broker1

- Execution
- Affirmation
- Confirmation
- Transfer
- Portfolio

Filter

Consumed Unconsumed

Upload

row	executionType	executionVenue	price_accruedInterest	quantity_amount	linearId
0	ELECTRONIC	Execution Venue	1.3700	800000.00	8f8fb522-9d11-48b2-9e6c-c53..
1	ELECTRONIC	Execution Venue	1.3700	800000.00	1f517b62-4f35-4877-a603-5d3.
2	ELECTRONIC	Execution Venue	1.3700	800000.00	9d13a93a-fb28-444b-af4f-c998
3	ELECTRONIC	Execution Venue	1.3700	800000.00	a6c0f89d-d271-4ce0-aef5-c237
4	ELECTRONIC	Execution Venue	1.3700	800000.00	57bc3fb9-0e7d-4275-b10c-07f.
5	ELECTRONIC	Execution Venue	1.3700	800000.00	7e9f5167-1b72-407e-b1c0-e11.
6	ELECTRONIC	Execution Venue	1.3700	800000.00	1b413e90-e4e2-4f8d-b7c6-777.
7	ELECTRONIC	Execution Venue	1.3700	800000.00	96d7b339-45c6-45cc-a79b-59a
8	ELECTRONIC	Execution Venue	1.0400	300000.00	5ca7c295-8209-47f3-af73-5323
9	ELECTRONIC	Execution Venue	1.3700	800000.00	97fb1ec8-aea4-4207-9999-25b.
10	ELECTRONIC	Execution Venue	1.3700	320000.00	00a5dcd3-0ae6-4e51-a39e-975
11	ELECTRONIC	Execution Venue	1.3700	480000.00	2287ac24-9b80-40a6-89c4-1ff0
12	ELECTRONIC	Execution Venue	1.3700	320000.00	3bf6d813-e1e8-4507-8127-28e
13	ELECTRONIC	Execution Venue	1.3700	480000.00	df521ba3-c6f5-4dbf-8aa6-8625

Vault Level Front End

exactpro
EXITUS ACTA PROBAT

Barclays Derivhack - Exactpro

broker1

- Execution
- Affirmation
- Confirmation
- Transfer
- Portfolio

Filter

Consumed Unconsumed

Upload

row	execution				linearId
0	ELECTRO				8f8fb522-9d11-48b2-9e6c-c53..
1	ELECTRO				1f517b62-4f35-4877-a603-5d3.
2	ELECTRO				9d13a93a-fb28-444b-af4f-c998
3	ELECTRO				a6c0f89d-d271-4ce0-aef5-c237
4	ELECTRO				57bc3fb9-0e7d-4275-b10c-07f.
5	ELECTRO				7e9f5167-1b72-407e-b1c0-e11.
6	ELECTRO				1b413e90-e4e2-4f8d-b7c6-777.
7	ELECTRO				96d7b339-45c6-45cc-a79b-59a
8	ELECTRO				5ca7c295-8209-47f3-af73-5323
9	ELECTRO				97fb1ec8-aea4-4207-9999-25b.
10	ELECTRO				00a5dcd3-0ae6-4e51-a39e-975
11	ELECTRO				2287ac24-9b80-40a6-89c4-1ff0
12	ELECTRO				3bf6d813-e1e8-4507-8127-28e
13	ELECTRONIC	Execution Venue	1.3700	480000.00	df521ba3-c6f5-4dbf-8aa6-8625

helidon SE

TypeScript

React

ag Grid

Use Case 1: Execution

exactpro
EXITUS ACTA PROBAT

Barclays Derivhack - Exactpro

broker1

- Execution
- Affirmation
- Confirmation
- Transfer
- Portfolio

Filter

Consumed Unconsumed

Upload Execution

row	executionType	executionVenue	price_accruedInterest	quantity_amount	linearId	Confirmation
0	ELECTRONIC	Execution Venue	1.3700	800000.00	71f01a90-f9	Instructions
1	ELECTRONIC	Execution Venue	1.3700	800000.00	7db0c018-d	Instructions
2	ELECTRONIC	Execution Venue	1.3700	800000.00	434a9169-7	Instructions

exactpro
EXITUS ACTA PROBAT

Chatbot Use Cases

Conversation

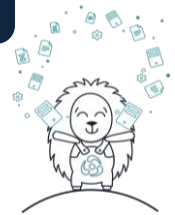
Generic

Personalized

Actions

API, transfer to Operator

API, transfer to Operator



information requests

action requests

Vault Queries

Flows Initiation

r3

corda enterprise