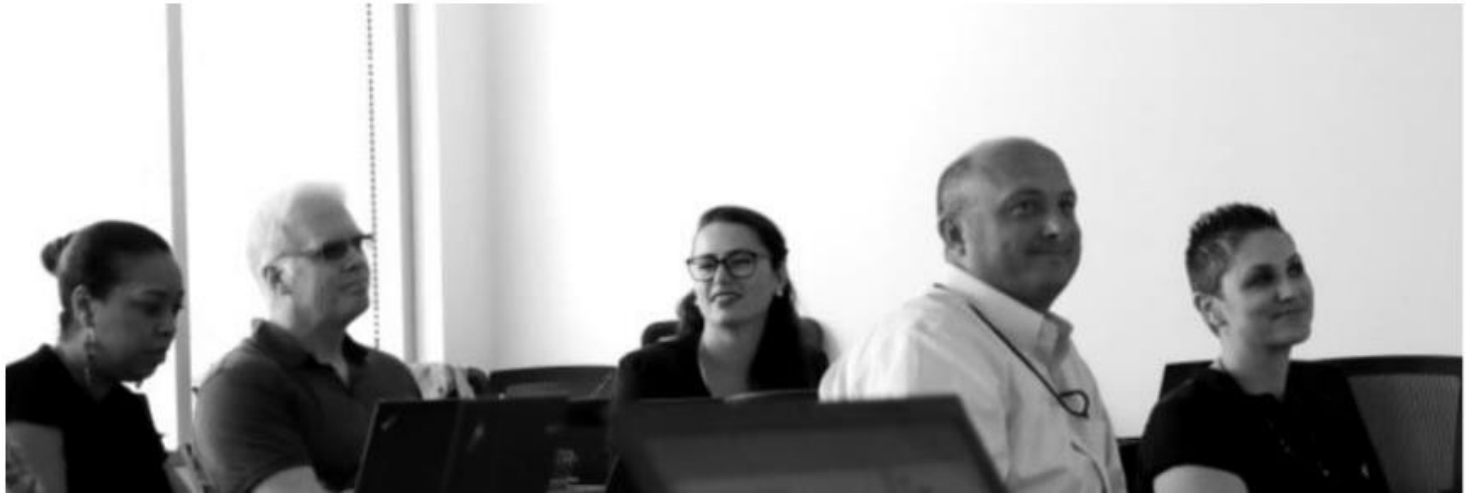
CONCEPTS OF REVISIONING

Source : Guidewire Educational Platform

Concepts of Revisioning



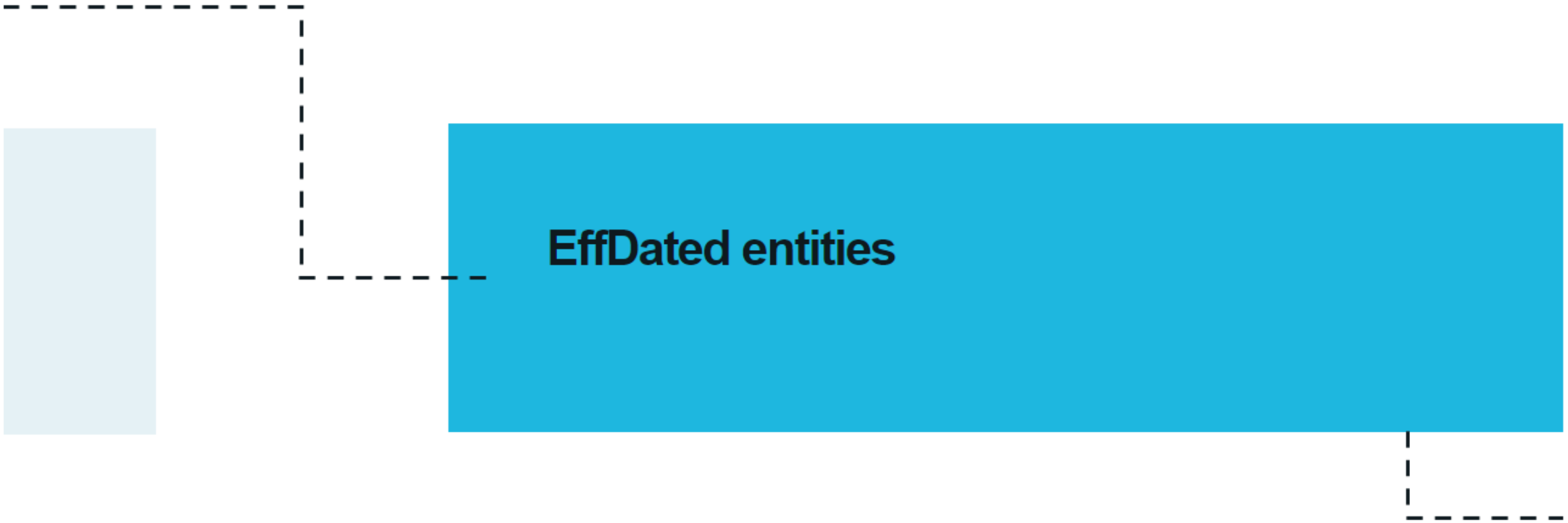
Source : Guidewire Educational Platform

Lesson objectives

By the end of this lesson, you will be able to:

- Describe policy revisioning and EffDated entities
- Describe window mode and slice mode
- Access a policy in window and slice mode using Gosu

Source : Guidewire Educational Platform

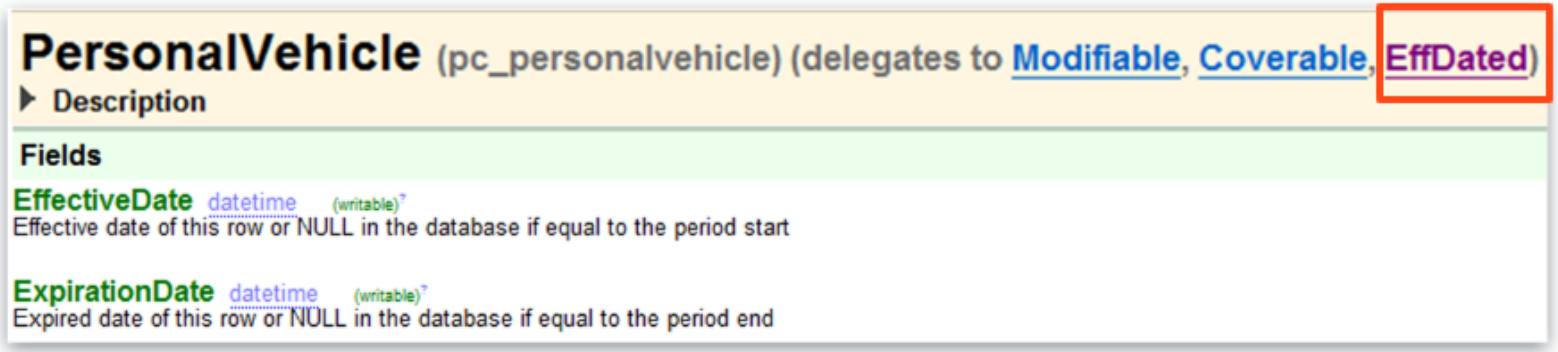


Source : Guidewire Educational Platform

EffDated entities

Are created to track the state of an entity over effective time

- Have EffectiveDate and ExpirationDate columns
- Are members of an effective dated graph, rooted at an effdatedbranch entity.



Source : Guidewire Educational Platform

PolicyPeriod entity

- PolicyPeriod entity is:
  - the root of all EffDated entities on a policy graph
  - **not** EffDated, but type of **effDatedBranch**
- Set relationship between an EffDated entity and PolicyPeriod
  - effDatedBranchType="PolicyPeriod" in the eti file

PersonalVehicle.eti

```
1  <?xml version="1.0"?>
2  <entity
3    xmlns="http://guidewire.com/datamodel"
4    desc="Personal Vehicle"
5    displayName="Personal Vehicle"
6    effDatedBranchType="PolicyPeriod"
7    entity="PersonalVehicle"
```

Source : Guidewire Educational Platform

EffDated entity columns

- **EffectiveDate** and **ExpirationDate**
  - *null* value means this entity has same effective and expiration dates as its root (i.e. PolicyPeriod.PeriodStart and PolicyPeriod.PeriodEnd)
- **FixedId** – a unique identifier that PolicyCenter generates when a new object is created
  - Cannot be null
  - Stays same for all version of the object
- **Branch** – link to the associated PolicyPeriod
  - Cannot be null
- **BasedOn** – a pointer back to prior version of this entity
  - If this is the first version, BasedOn is null
  - Column on EffDated entity as well as the PolicyPeriod

Source : Guidewire Educational Platform

Creating an EffDated entity

- First, identify the root entity of type EffDatedBranch, such as PolicyPeriod
- Next , create the EffDated entity
  - Set the entity attributes
    - type=”EffDated”
    - effDatedBranchType="PolicyPeriod“
  - Follow other rules relevant to the entity (e.g., PolicyLine or Coverage entities have their own specific rules)
- Refer to the existing EffDated entities as examples
  - For example, PersonalVehicle.eti, PersonalAutoCov.eti, BOPBuilding.eti, etc.

Source : Guidewire Educational Platform

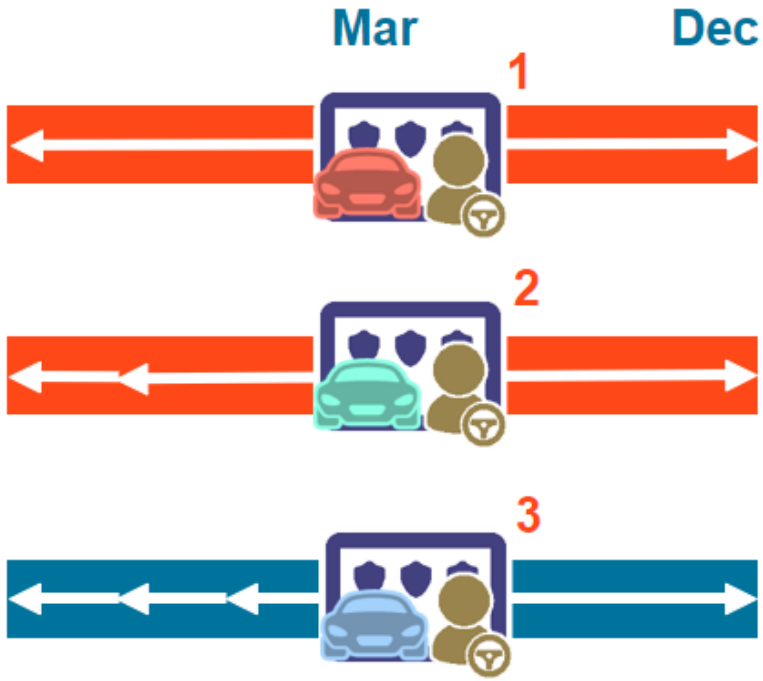
BasedOn example

Driver and PersonalVehicle EffDated entities have BasedOn properties

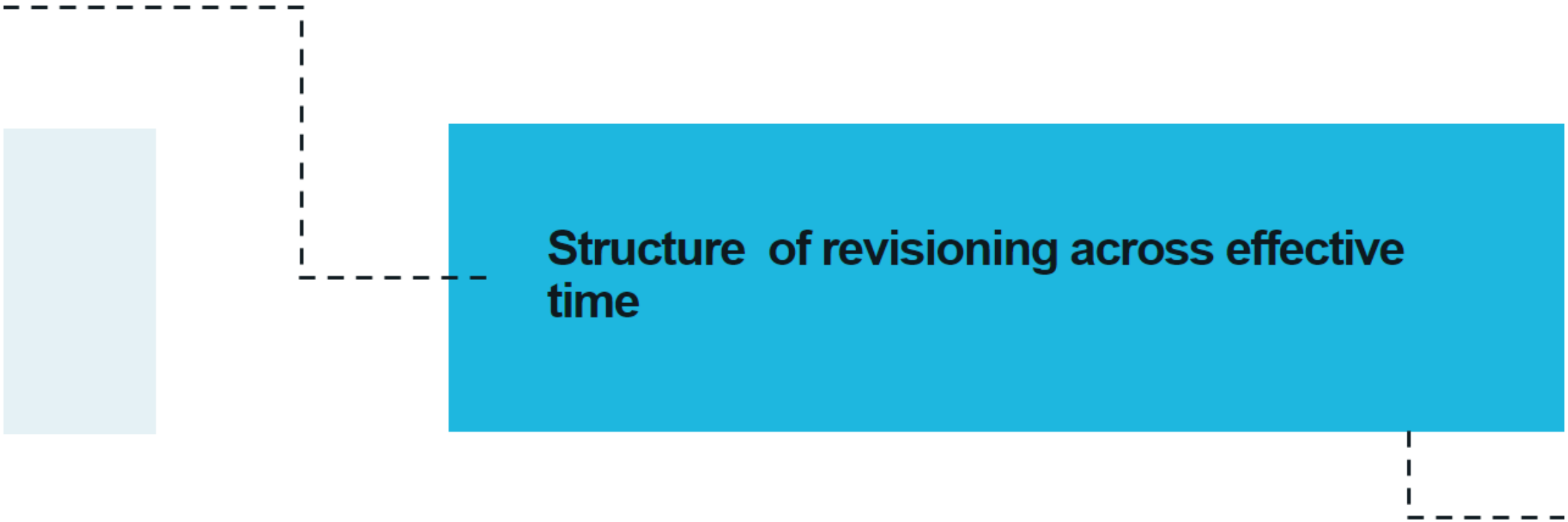
Submission on Mar 1 (Eff Mar 1 – Dec 31)
policy with 1 car and 1 driver
DriverV1.BasedOn = null
PersonalVehicleV1.BasedOn = null

Policy Change 1 on Mar 2 (Eff Apr 1 - Dec 31)
change car 1, no change in driver
DriverV2.BasedOn = DriverV1
PersonalVehicleV2.BasedOn = PersonalVehicleV1

Policy Change 2 on Mar 3 – (Eff May 1 - Dec 31)
change car 1, no change in driver
DriverV3.BasedOn = DriverV2
PersonalVehicleV3.BasedOn = PersonalVehicleV2



Source : Guidewire Educational Platform

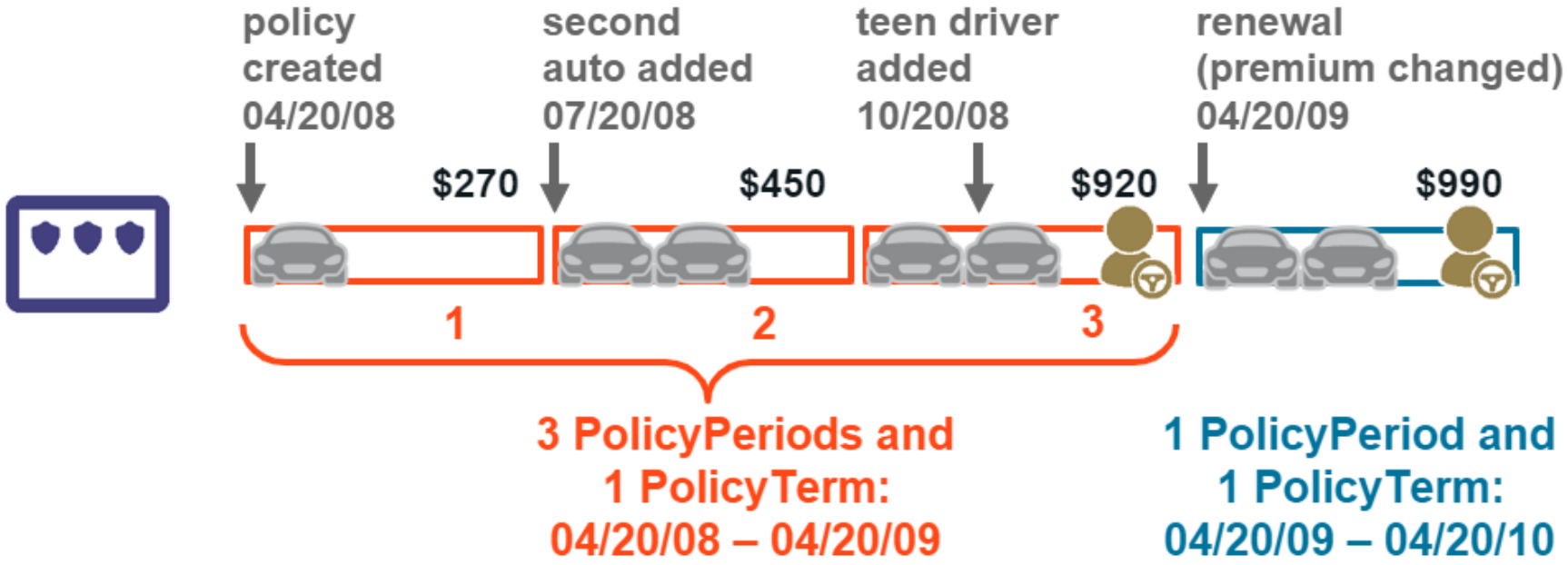


Source : Guidewire Educational Platform

PolicyPeriod

A PolicyPeriod is one "version" of the policy

- Has effective and expiration dates (PeriodStart and PeriodEnd) which define the actual term of the policy
- Has an EditEffectiveDate, which is the date at which changes from a job are applied



Source : Guidewire Educational Platform

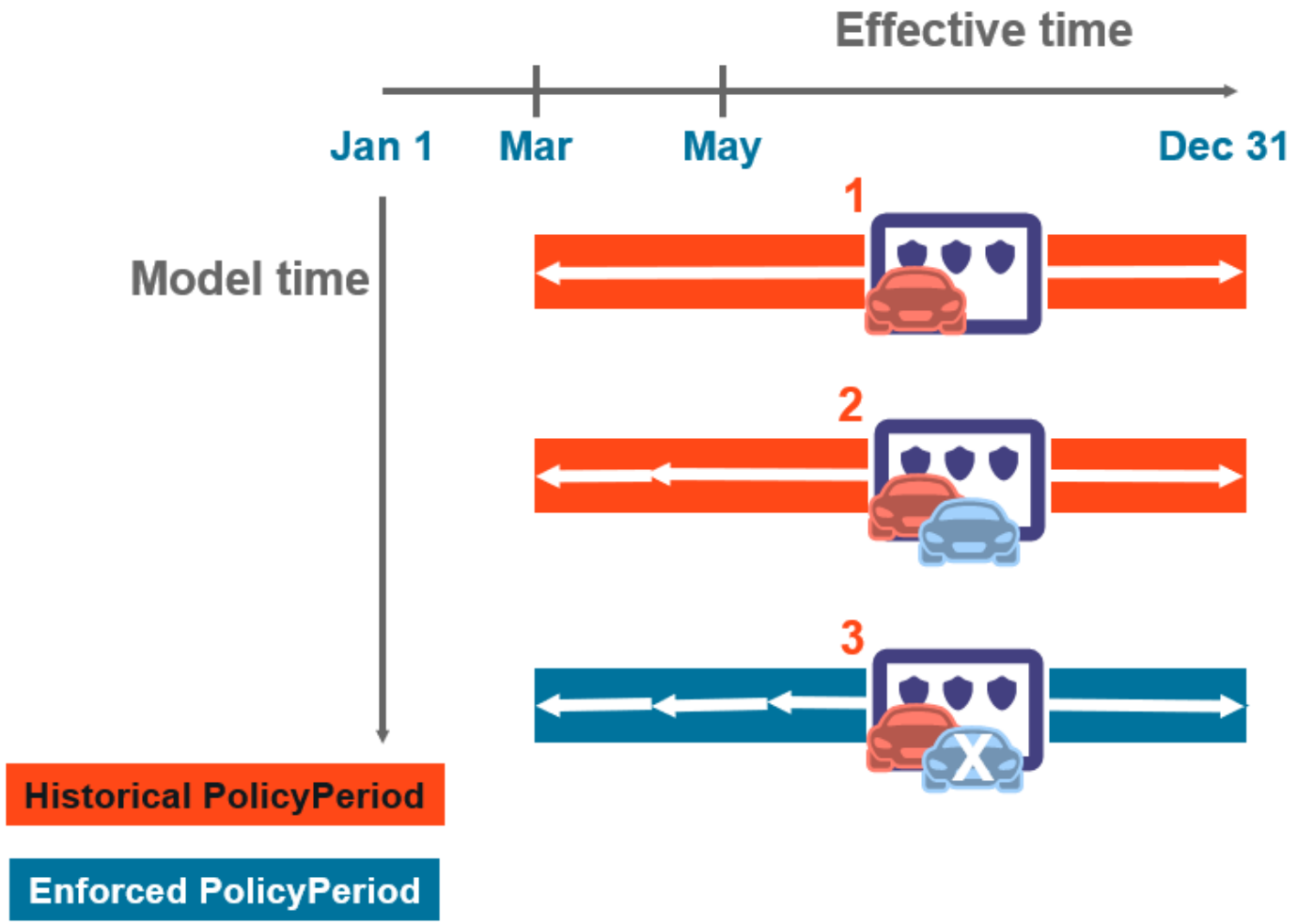
Model time and effective time

To track policy changes over time, a policy must be considered in two different time dimensions:

- **Model time** is the actual real-world time when policies are created or jobs are bound
- **Effective time** is the time dimension of the policy itself within the policy period

Source : Guidewire Educational Platform

Example



Mar 1 – policy with 1 car
effective Apr 1 - Dec 31
Model date: Mar 1
Effective date: Apr 1

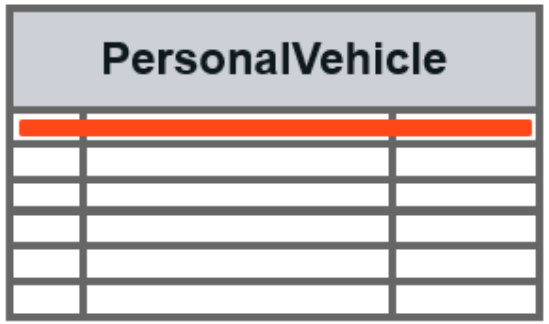
Mar 2 – add car 2
effective May 1 - Dec 31
Model date: Mar 2
Effective date: May 1

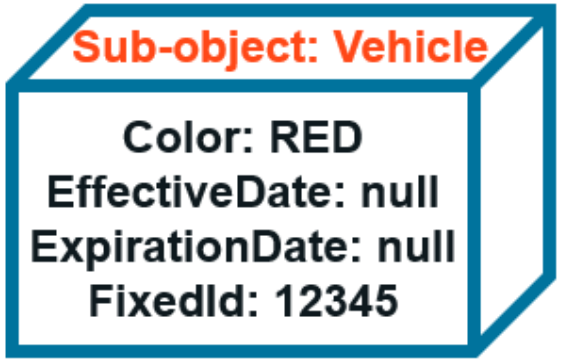
Mar 3 – Remove car 2
effective May 1 - Dec 31
Model date: Mar 3
Effective date: May 1

Source : Guidewire Educational Platform

Sub-objects across effective time (1)

- Every policy revision is the root of a complex graph of sub-objects such as drivers, vehicles, coverages, and so on
  - If one of these objects changes then that object has its own effective and expiration date
- Consider an example, where a personal auto submission is created with one vehicle





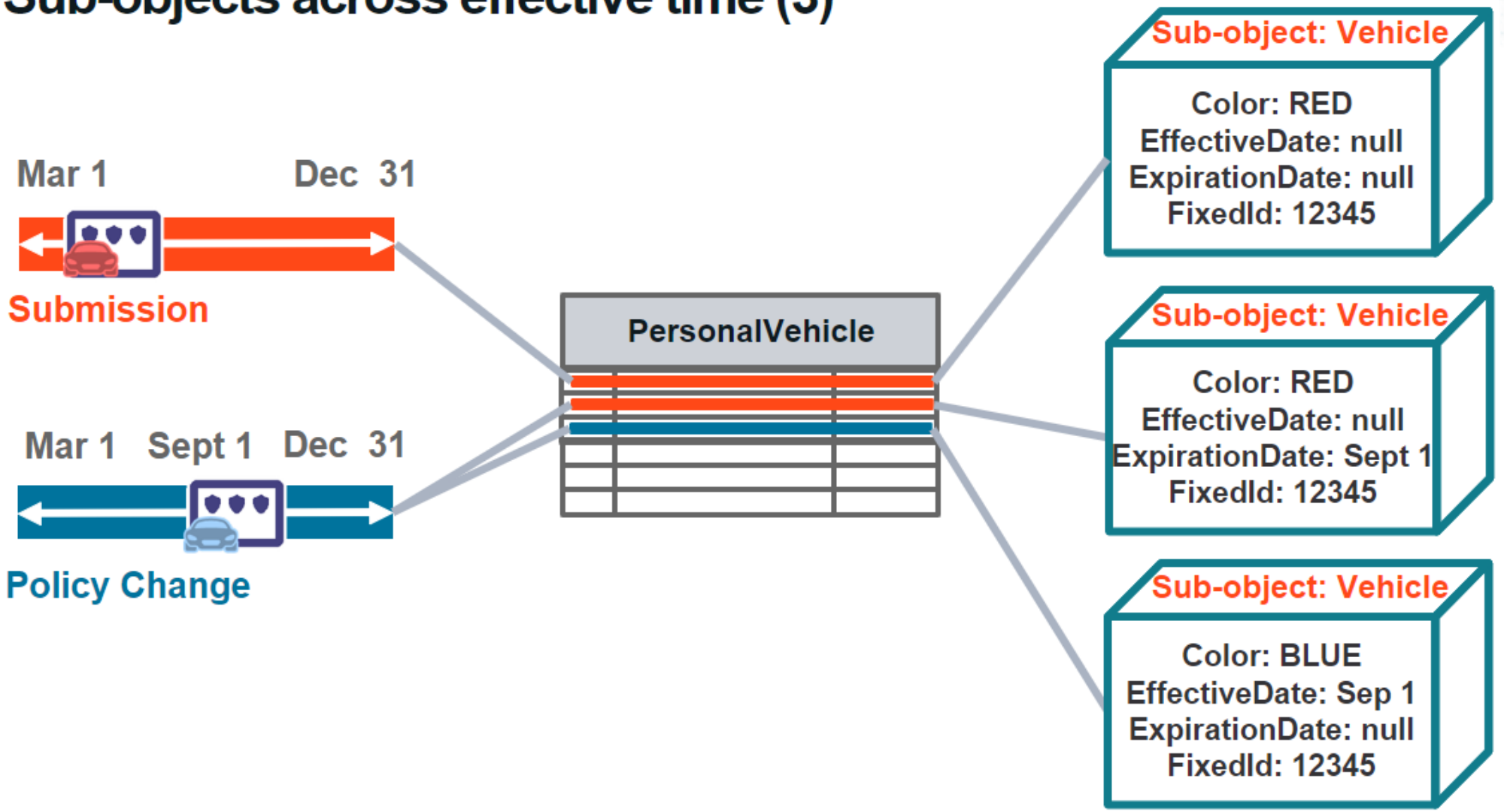
Source : Guidewire Educational Platform

Sub-objects across effective time (2)

- On Sep 1, the customer calls and says the car was painted blue today
- A new policy revision is created by cloning a new row in the database differing in effective date and expiration date
- Expiration date of first row is the effective date of second row
- A row is effective at a date specified if:
  - Effective date <= specified date < expiration date

Source : Guidewire Educational Platform

Sub-objects across effective time (3)



Source : Guidewire Educational Platform

Window mode and slice mode overview

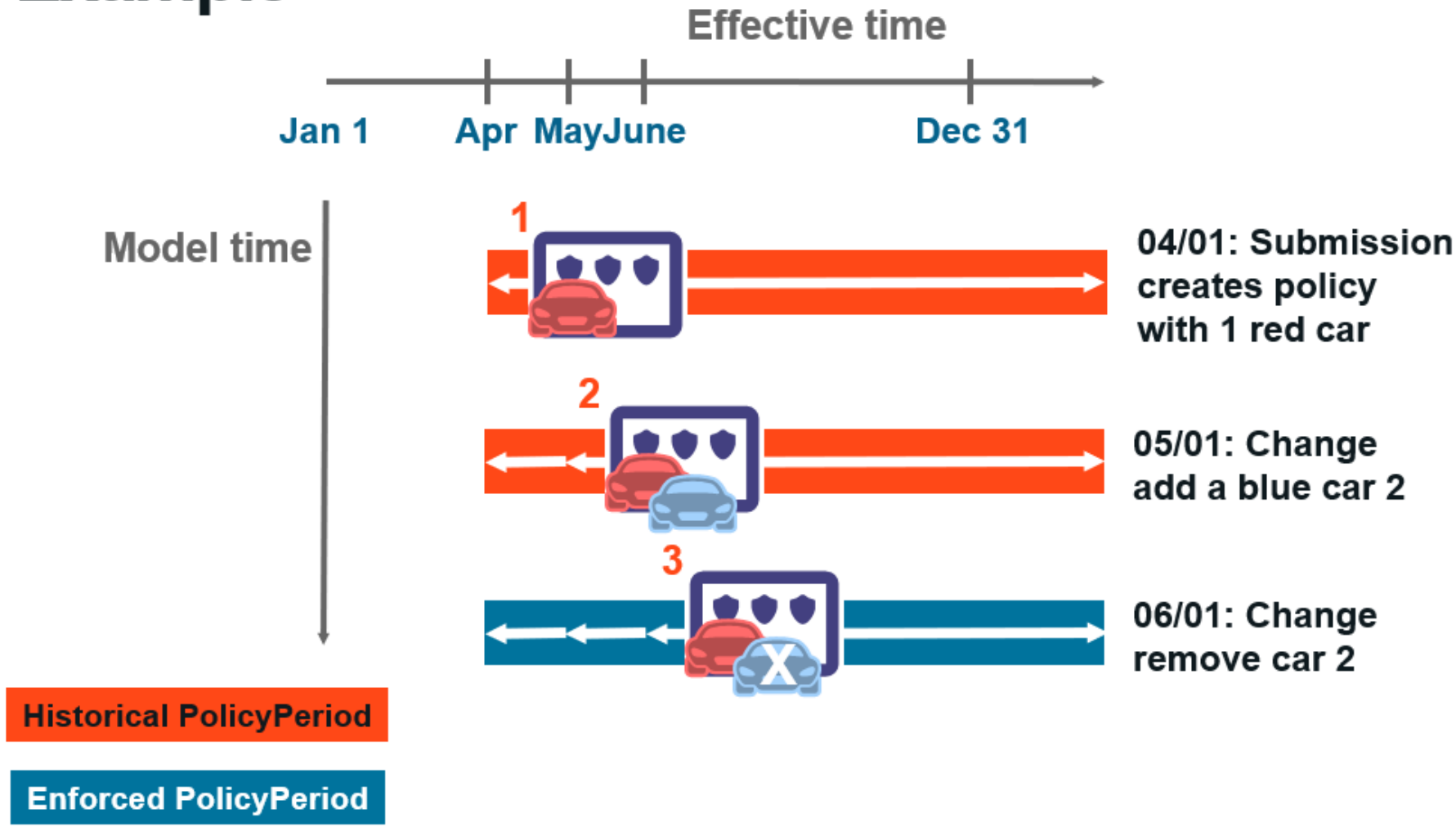
Source : Guidewire Educational Platform

Two modes to access a branch's objects

- Usually, jobs change policy data across effective time
- Slice mode is viewing policy data as of a specific date in effective time
  - When you request objects on the policy in slice mode, PolicyCenter automatically gets the correct version of each object as of the slice date
- Window mode is viewing all versions of policy data across all dates in effective time
  - For example, you want to get all the costs for a vehicle within a policy term (cost may vary if coverages have changed)
  - Or you want to list all the drivers ever assigned to a vehicle (a driver may only be assigned for short period)

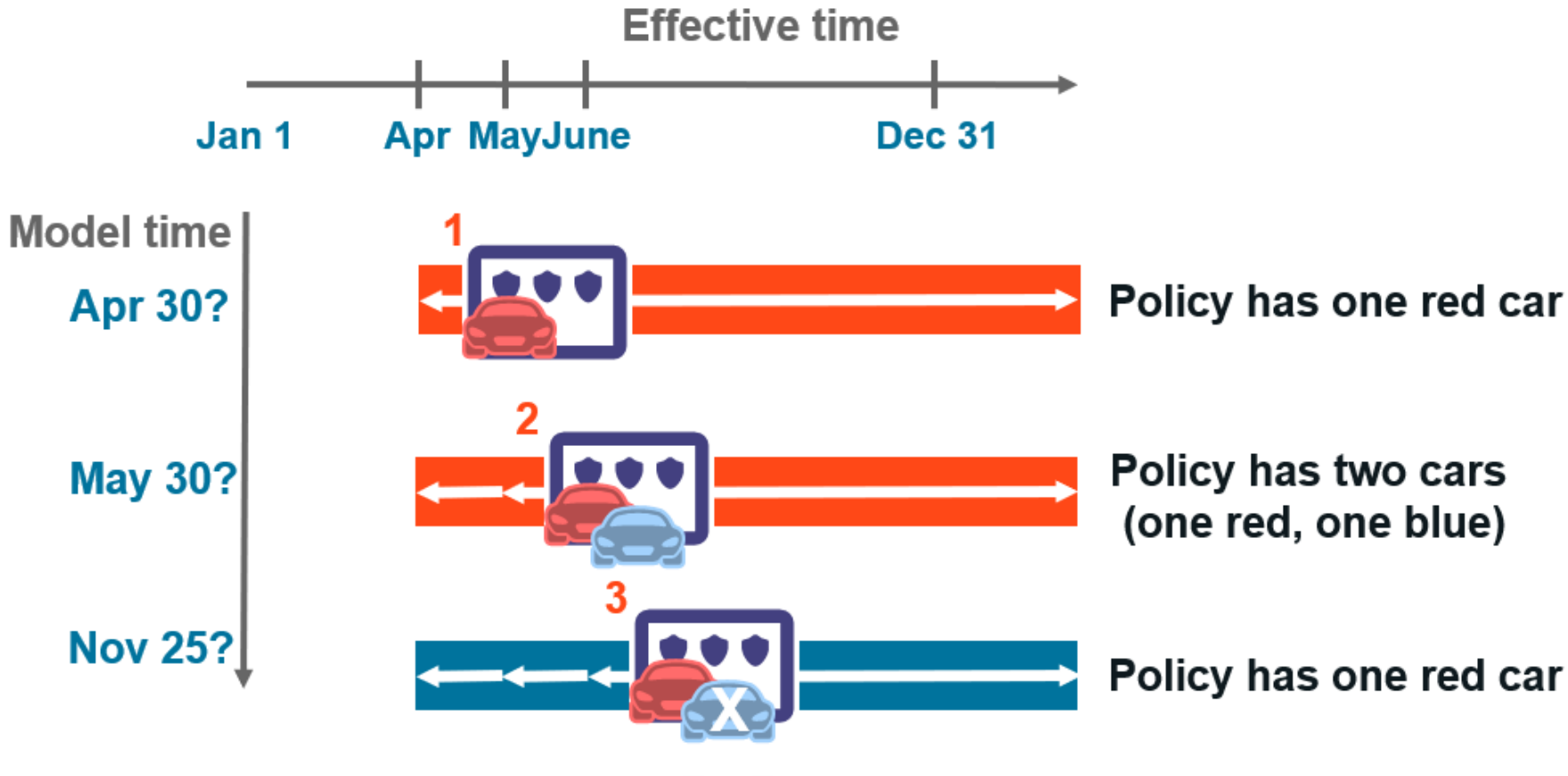
Source : Guidewire Educational Platform

Example



Source : Guidewire Educational Platform

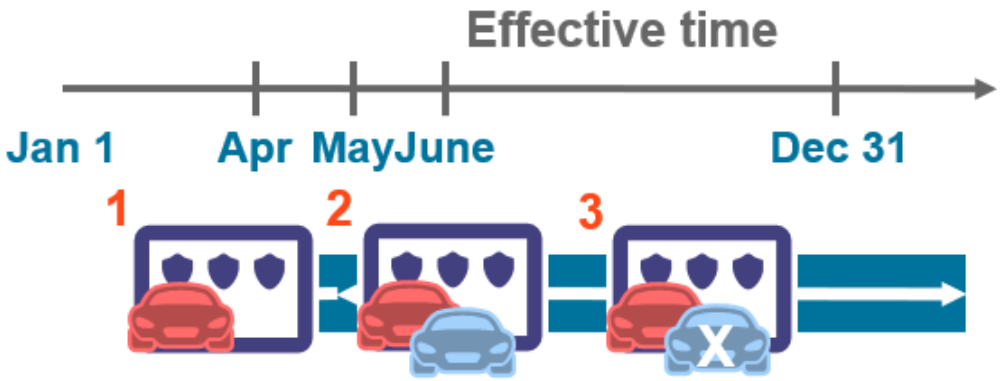
Slice mode view as of:



On any date, you might not see all cars that exist at some point in that policy period

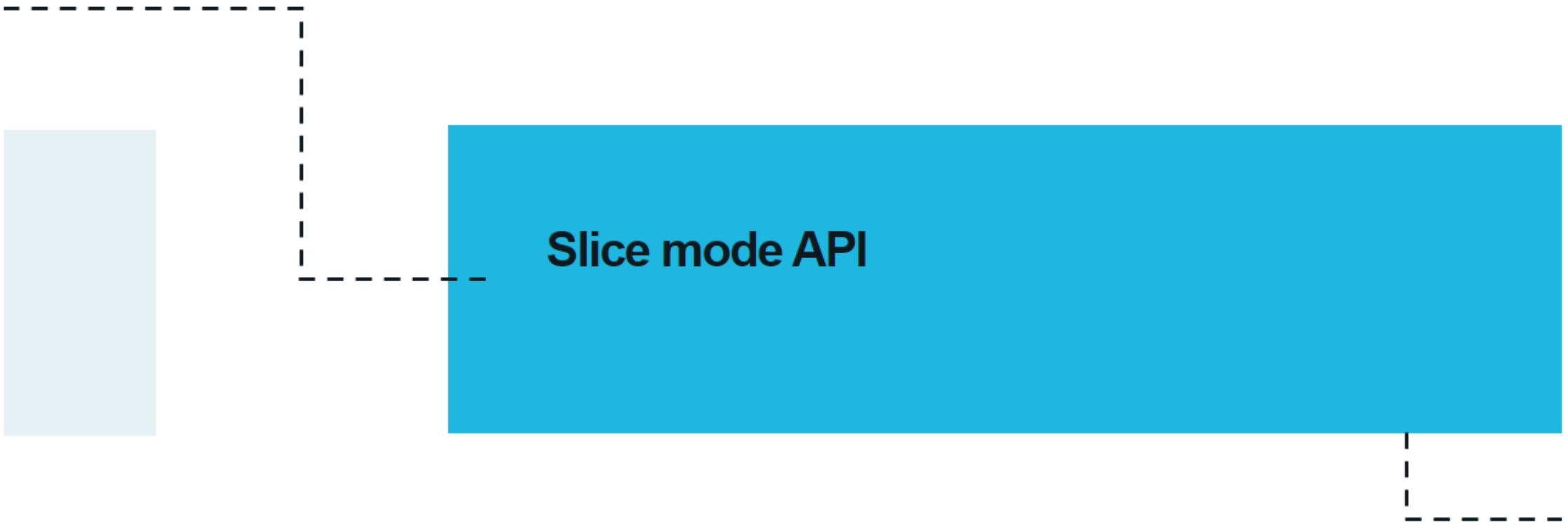
Source : Guidewire Educational Platform

Window mode view



- Lists all cars that existed on the policy for any period of time with the effective dates
  - First, get a list of all cars that were ever on the policy
  - Next, for each car, you want to iterate across all changes to that car across effective time
  - Obtained by using the VersionList property on a policy period
- There are two cars in VersionList but there are three versions of the policy
  - Submission, Policy Change 1, Policy Change 2

Source : Guidewire Educational Platform



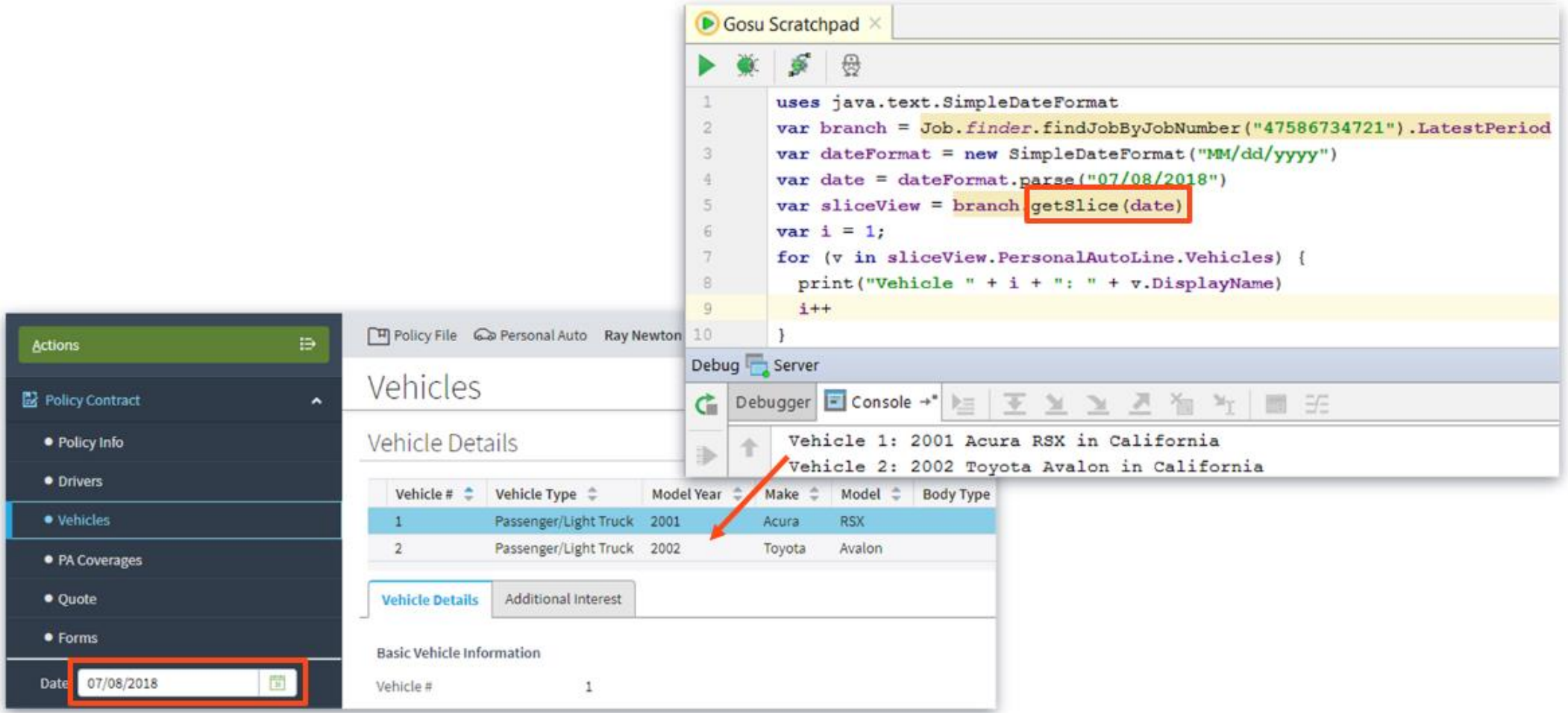
Source : Guidewire Educational Platform

Slice mode API: getSlice(sliceDate)

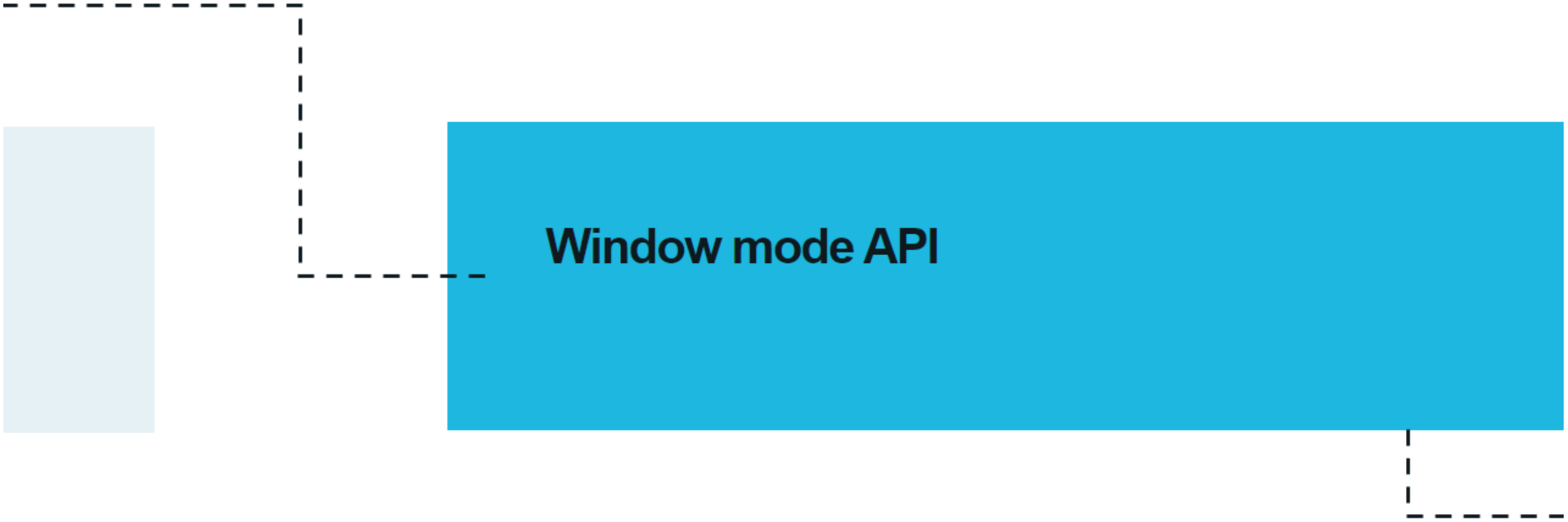
- Access a slice from a PolicyPeriod
- Syntax: ***slicedPolicyPeriod* = *aBranch*.getSlice(*sliceDate*)**
  - Return value is in slice mode
  - Returns sliced PolicyPeriod
  - Displays view of PolicyPeriod as of slice date

Source : Guidewire Educational Platform

Example: getSlice using Gosu Tester and verified using date selector in the UI



Source : Guidewire Educational Platform



Source : Guidewire Educational Platform

Window mode API: VersionList

- Version list represents all versions of an object in a policy period across effective time
- Use any object's **VersionList** property to get a version list
- **versionList.AllVersions** property gets all versions of this object
  - Sorted on effective date for each version
- **versionList.AsOf(date)** method gets the one version of this object on that date, or null if none were effective on that date

Source : Guidewire Educational Platform

Version list examples

- Get all versions of a particular car in a policy period across effective time :
  - var val = vehicleVL.AllVersions
    - If the car was added and then removed in a subsequent change transaction (as discussed previously), the car would still be listed
    - It existed on the policy period at some point
- Get the car version that is effective on ‘date’
  - var vehOnDate = vehicleVL.AsOf(date)
  - Returns
    - Single car object in window mode
    - Null if no version of the car is effective at that date

Source : Guidewire Educational Platform

VersionList examples: Get vehicles on a policy

- All versions of all vehicles that ever existed on the Policy

Gosu Source

```
1 // Get all versions of all vehicles that ever existed on a policy
2 var branch = Job.finder.findJobByJobNumber("206921").LatestPeriod
3 branch.PersonalAutoLine.VersionList.Vehicles.flatMap(λ vl → vl.AllVersions).each(λ v → print(v))
```

Runtime Output

```
1 2001 Mazda MPV in California
2 2003 Chevrolet Suburban in California
```

- All vehicles as of specific date

Gosu Source

```
1 // Get all vehicles as of a date
2 var branch = Job.finder.findJobByJobNumber("206921").LatestPeriod
3 branch.PersonalAutoLine.VersionList.VehiclesAsOf("2012-06-02").each(λ v → print(v))
```

Runtime Output

```
1 2001 Mazda MPV in California
```

Source : Guidewire Educational Platform

VersionList examples: Get costs on a policy

- All costs on a personal auto policy

Gosu Source

```
var branch = Job.finder.findJobByJobNumber("206921").LatestPeriod
branch.PersonalAutoLine.VersionList.PACosts.flatMap(λ cl → cl.AllVersions).each(λ c → print(c))
```

Runtime Output

```
Uninsured Motorist - Bodily Injury coverage on 2001 Mazda MPV in California
Medical Payments coverage on 2001 Mazda MPV in California
Uninsured Motorist - Property Damage coverage on 2003 Chevrolet Suburban in California
Uninsured Motorist - Property Damage coverage on 2001 Mazda MPV in California
Comprehensive coverage on 2001 Mazda MPV in California
Collision coverage on 2001 Mazda MPV in California
Liability - Bodily Injury and Property Damage coverage on 2001 Mazda MPV in California
CA Tax
Medical Payments coverage on 2003 Chevrolet Suburban in California
Liability - Bodily Injury and Property Damage coverage on 2003 Chevrolet Suburban in California
Uninsured Motorist - Bodily Injury coverage on 2003 Chevrolet Suburban in California
```

- All costs on a vehicle at a specific date

Gosu Source

```
var branch = Job.finder.findJobByJobNumber("206921").LatestPeriod
branch.PersonalAutoLine.VersionList.PACostsAsOf("2012-04-02").each( λc → print(c))
```

Source : Guidewire Educational Platform

Lesson objectives review

You are now able to:

- Describe policy revisioning and EffDated entities
- Describe window mode and slice mode
- Access a policy in window and slice mode using Gosu

Source : Guidewire Educational Platform

Capgemini