Chubb Enterprise Ontology · 2.0.0-draft · 2026-10-26

Enterprise Claim Ontology

Subject-matter expert review and consent pack

For: claims subject-matter experts in every department (Property, Casualty, Workers' Compensation, A&H, Marine, Specialty…) who will build their claim adjudication on the enterprise claim ontology.

What we ask: read each page, check that the picture and the words match how your business works, and tick *I agree*, *I agree with changes*, *I do not agree* or *Not my area*. Write any change in the box. Section 4 lists the decisions only you can make.

Time needed: about 2 minutes per page; 21 pages to review plus the decisions list.

The pictures are generated from the ontology itself, so what you approve is exactly what is built. Technical names appear only in small grey text, for traceability. The ontology team's version of the same pictures is the Ontology Specialist Guide.

1. What you are approving

The claims ontology (CLM) has been rebuilt as an **enterprise ontology**: one shared description of claims that every department uses, instead of each system describing claims its own way. It has three shared layers that every claim department imports, and each department adds its own terms in its own module on top:

- **Enterprise Foundation** – the thirteen basic kinds of thing (request, event, party, role, agreement…).
- **Enterprise Insurance Common** – the claim and its components, policy and coverage, parties and their roles, reserves, payments and recoveries, documents, insured objects. Shared with Underwriting and Finance.
- **Enterprise Claims domain** – the shared claim-handling framework: guided adjudication, coverage determination, assessments and valuations, investigation and fraud, negotiation and settlement, litigation, tasks and assignments.
- **Your department module** – your own kinds of claim component, roles, valuations, rule sets and code values (page C10).

1,949	**70**	**428**	**76**
terms from today's CLM model fixed or already right	applied, waiting for your wording or values	applied, still to be fixed (section 4)	business questions the model is tested against

2. How to read the pictures

Enterprise Foundation · Enterprise Insurance Common · Enterprise Claims domain · Underwriting · Code list · W3C / OMG standard · Department extension (example) · Missing link (open item)

Solid arrow = a relationship (read it as a sentence: *Claim → has claim component → Claim Component*). Dotted arrow with a circle = 'is a kind of'. Dotted arrow to a yellow pill = 'takes its value from this code list'. Red dashed = a link that is still missing.

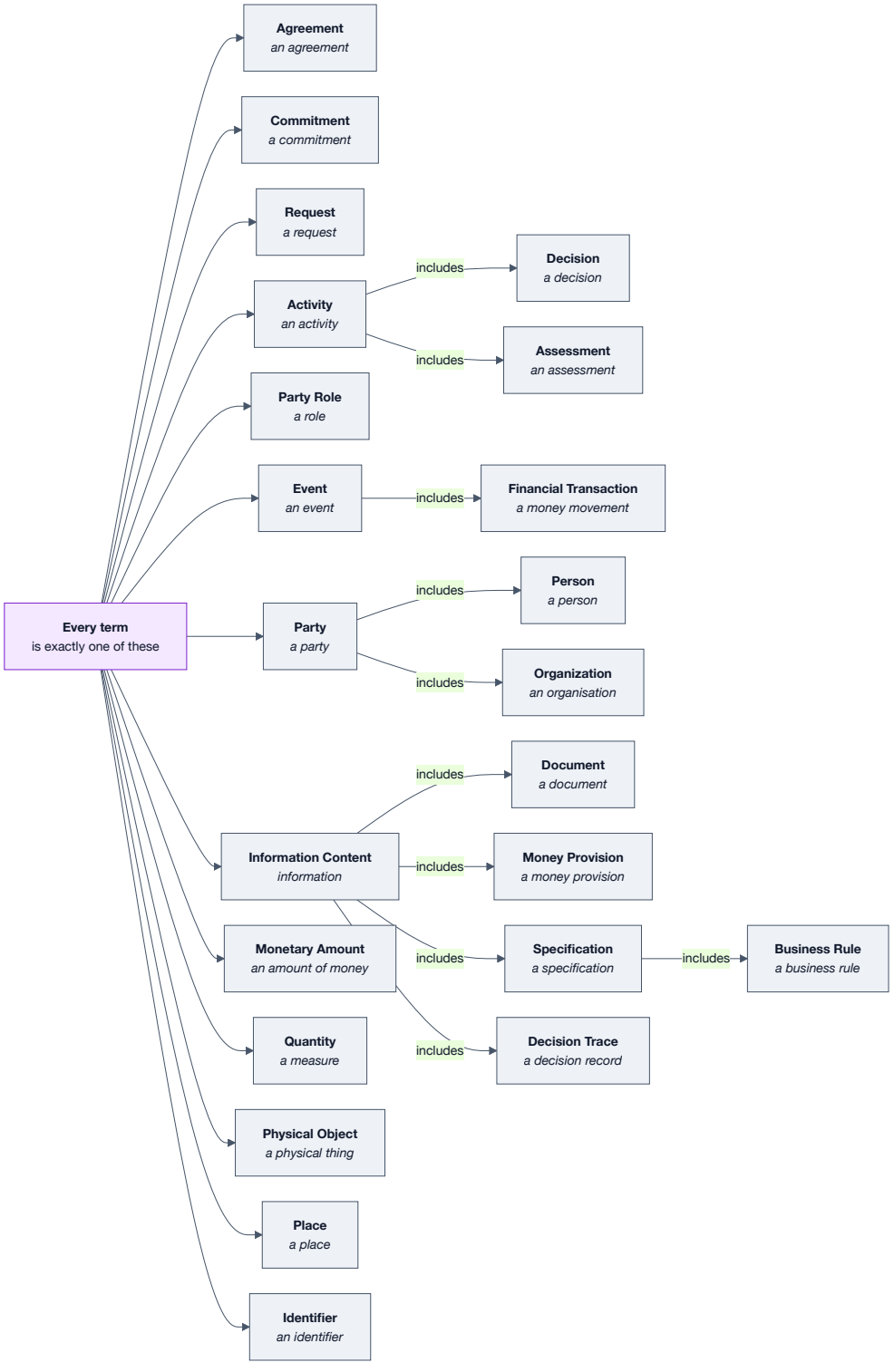
Read every solid arrow as a sentence, from the box it starts at to the box it points to. The small italic line in each box says what kind of thing it is (a request, an event, a role…).

3. Pages to review

- A01 – Everything is exactly one kind of thing
- A02 – A claim is a request for benefits
- A03 – People and companies play roles on a claim
- A04 – The policy promises coverage; limits and deductibles qualify it
- A05 – Money is an amount with a currency; reserves are estimates, payments are events
- A06 – Every decision records its outcome, its reason and how it was reached
- A07 – Assessments find the facts; valuations put a price on them
- A08 – Codes come from shared, managed code lists
- A09 – Shared kinds of link let every department ask the same question
- A10 – How dates are checked
- A11 – Every automated result says where it came from
- C01 – The claim and what it is about
- C02 – Reserves, payments and recoveries
- C03 – How a claim is adjudicated
- C04 – Evaluating the loss and settling it

- C05 – Investigating the claim and handling suspected fraud
- C06 – Legal actions on a claim
- C07 – The things that were damaged
- C08 – Documents, notes and correspondence
- C09 – Who handles the claim and what they must do
- C10 – How your department adds its own terms (example)

A01 – Everything is exactly one kind of thing



A01 – Everything is exactly one kind of thing – read left to right: every term is exactly one of the thirteen basic kinds, some of which include narrower kinds.

What this says. Every term in the enterprise ontology is ultimately one of thirteen basic kinds of thing – an agreement, a commitment, a request, a party, a role, an event, an activity, information, an amount of money, a

measure, a physical thing, a place or an identifier – and never two of them at once. Decisions and assessments are activities; payments and postings are events; documents and reserves are information.

Why we chose it. One clear top level is what lets every department add its own terms without contradicting another department: a Property claim, a Casualty claim and an A&H claim all sit under the same 'request'. The thirteen roots follow FIBO, OMG Commons and gist; the 'never two at once' rule is the W3C OWL 2 disjointness axiom, which the reasoner (HermiT) checks – it found no term breaking it.

What changed from today's CLM model. The CLM ontology had nine roots, some overlapping (a claim was an activity and a request at once). Now each term has exactly one root.

Still to be fixed here

- **OI-08** Classes with no attributes or relationships of their own: `fnd:Identifier`

Please confirm

- Are these thirteen kinds of thing enough to place every concept your department uses?
- Is it acceptable that a reserve is treated as information (an estimate) rather than as money moved?

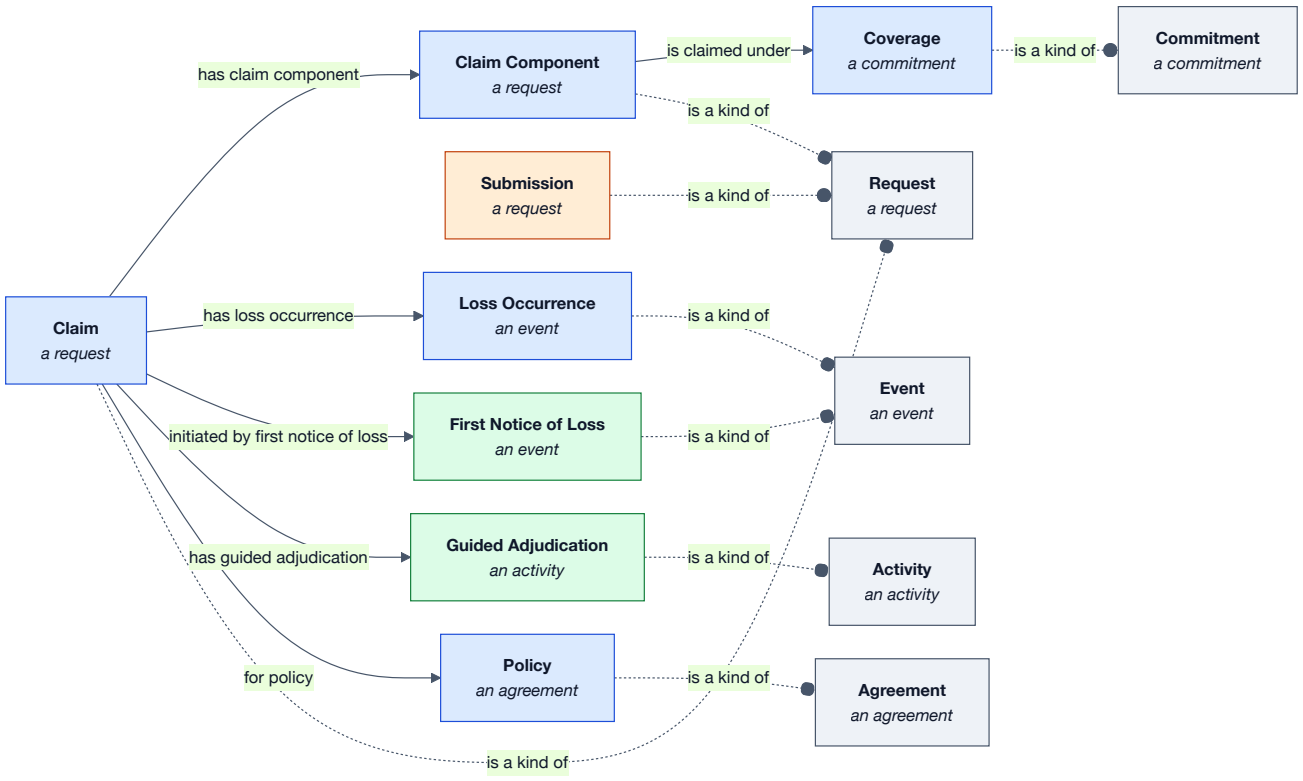
Your decision on A01 – Everything is exactly one kind of thing

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A02 – A claim is a request for benefits



A02 – A claim is a request for benefits

What this says. A claim is a request made under a policy for payment, defence or other benefits. It is split into claim components – the part of the claim for one claimant under one coverage. The accident or event that caused the loss is a separate thing (a loss occurrence), and so is the first notice of loss that reported it. Handling the claim (guided adjudication, investigation, negotiation) is an activity performed on the claim, not the claim itself.

Why we chose it. NAIC, the OMG P&C Data Model, Oracle OIDF and IBM IIW all define a claim as a request or demand; ACORD separates the claim from the claim activity. Keeping them apart lets one loss produce several claims, and one claim be handled by several activities, without mixing their dates and statuses.

What changed from today's CLM model. Claim Exposure, Claim Line and Claim Feature are now one concept, Claim Component (ACORD 'Claim Component', IBM 'Elementary Claim'), and the old Loss Event is the Loss Occurrence.

> **Still to be fixed here**
>
> - **OI-06** Relationships and attributes that say the same thing twice: `ins:forPolicy`
> - **OI-10** Facts the business questions ask for that the model cannot hold: `uw:Submission`

Please confirm

- Do 'claim component' and 'loss occurrence' match how your department talks about exposures and accidents?
- Is one claim component always one claimant under one coverage in your department?

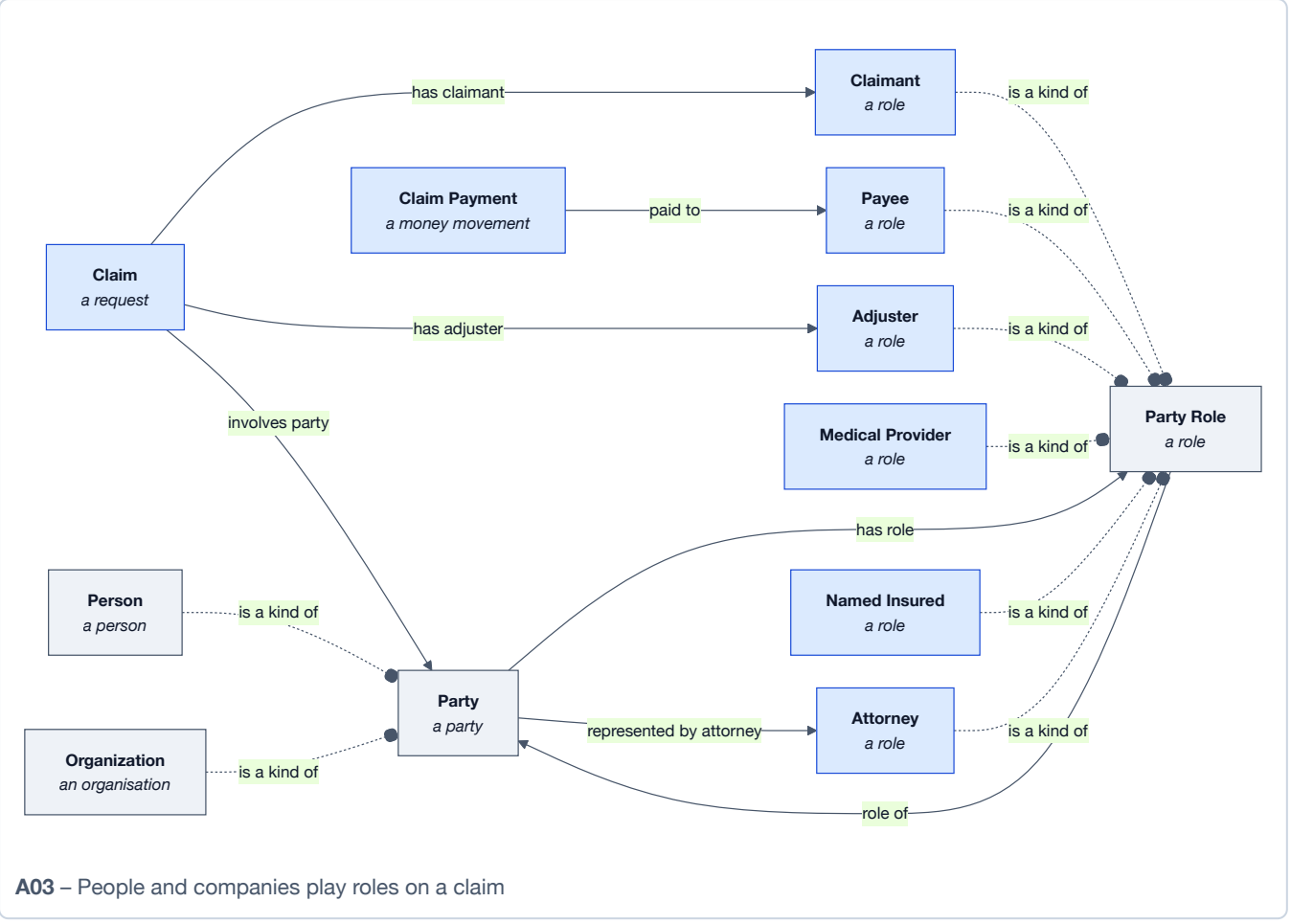
Your decision on A02 – A claim is a request for benefits

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A03 – People and companies play roles on a claim



A03 – People and companies play roles on a claim

What this says. A person or organisation is recorded once, as a party. What they are on a given claim – claimant, adjuster, attorney, medical provider, payee, named insured – is a role they play in that context. The same person can be a claimant on one claim and a witness on another without being stored twice.

Why we chose it. OMG Commons, the OMG P&C Data Model and ACORD all model party roles this way; OntoClean shows why a role (something you can stop being) must not be a kind of person. It is what lets Claims, Underwriting and Finance share one customer record.

What changed from today's CLM model. The CLM model had 'Participant' and several role classes that also carried name and address. Those attributes now live on the party; the roles carry only what belongs to the role.

Still to be fixed here

- **OI-06** Relationships and attributes that say the same thing twice: `ins:hasAdjuster`
- **OI-08** Classes with no attributes or relationships of their own: `ins:Claimant, ins:Payee`

Please confirm

- Are there roles your department needs that are not in the enterprise list (e.g. nurse case manager)?
- Should a payee always be a role of an existing party, or can it be a one-off bank account?

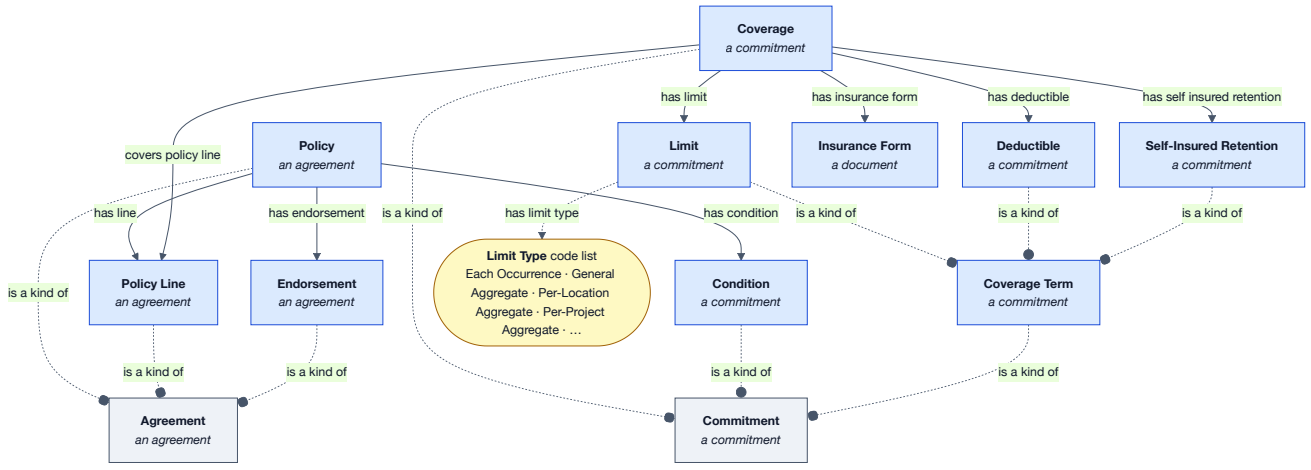
Your decision on A03 – People and companies play roles on a claim

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A04 – The policy promises coverage; limits and deductibles qualify it



A04 – The policy promises coverage; limits and deductibles qualify it

What this says. The policy is the agreement; each coverage is a promise (commitment) the insurer makes inside it, for one policy line. Limits, deductibles and self-insured retentions are coverage terms that qualify the promise, and the kind of limit (each occurrence, general aggregate…) is a value from a shared code list. Conditions are promises the insured must keep; endorsements are agreements that amend the policy.

Why we chose it. FIBO and gist separate the agreement from the commitments it creates; the OMG P&C model and IBM IIW model coverage as a component of the policy. Claims adjudicate against the coverage, so a claim component points to the coverage it is claimed under.

What changed from today's CLM model. Limits and deductibles were separate classes per limit kind in CLM; they are now one Limit (or Deductible) with a limit-type value. Some flat limit amounts on coverage and policy remain and are listed as an open item.

Still to be fixed here

- **OI-08** Classes with no attributes or relationships of their own: `ins:Condition`, `ins:SelfInsuredRetention`

Please confirm

- Can every limit your department uses be expressed as a Limit with a limit type?
- Are conditions and exclusions modelled at the right level (policy vs coverage) for your line?

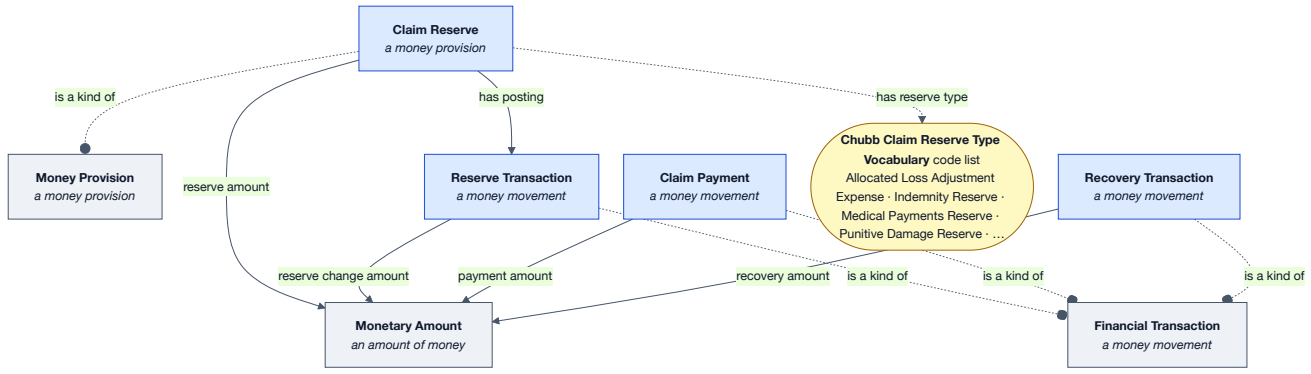
Your decision on A04 – The policy promises coverage; limits and deductibles qualify it

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A05 – Money is an amount with a currency; reserves are estimates, payments are events



A05 – Money is an amount with a currency; reserves are estimates, payments are events

What this says. Every amount of money is recorded with its currency. A reserve is the insurer's current estimate of what is still to be paid – information that changes over time – and each change is a dated reserve posting. Payments and recoveries are dated money movements; the kind of reserve, payment or recovery is a value from an enterprise code list.

Why we chose it. FIBO and ACORD always pair an amount with its currency; IBM IIW calls a reserve a 'Money Provision' and keeps it apart from the financial transactions that change it; NAIC SSAP 55 defines reserves as estimates.

What changed from today's CLM model. About 500 amounts that were plain numbers or text in CLM now point to a monetary amount with a currency. A few still do not (e.g. total incurred, total outstanding) and are in the open-items list.

Please confirm

- Is 'reserve = estimate (information), posting = event' how Finance and Claims see reserves?
- Which reserve, payment and recovery types must every department use, and which are department-only?

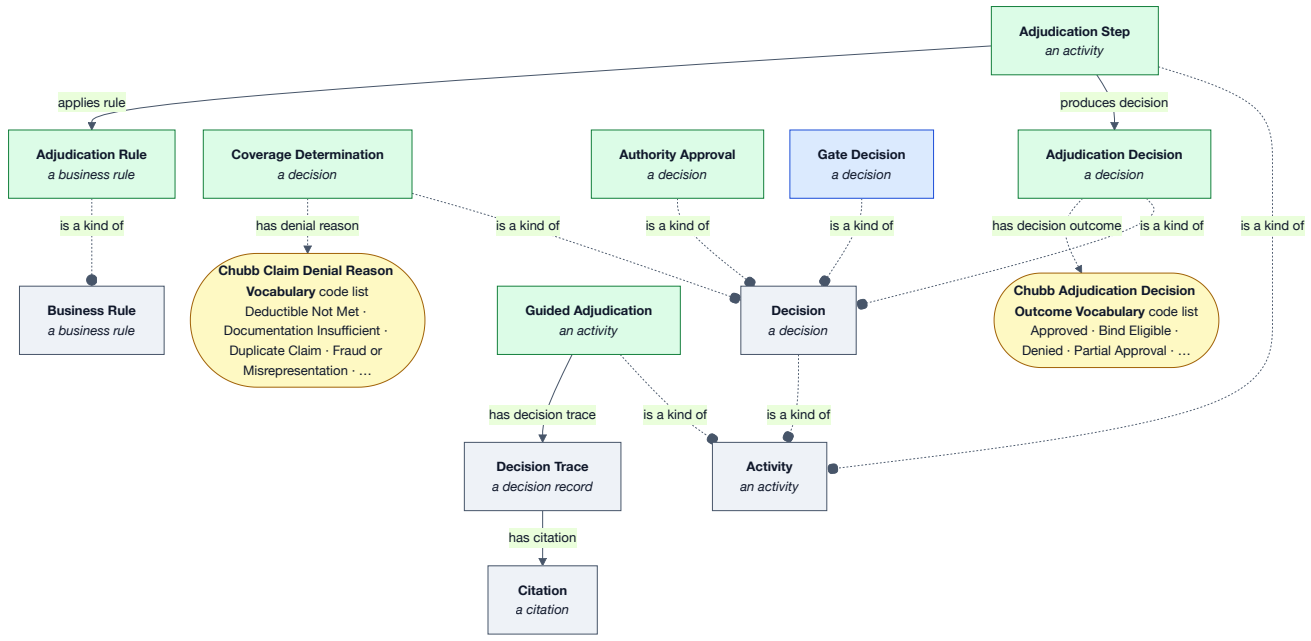
Your decision on A05 – Money is an amount with a currency; reserves are estimates, payments are events

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A06 – Every decision records its outcome, its reason and how it was reached



A06 – Every decision records its outcome, its reason and how it was reached

What this says. Every decision – an adjudication decision, a coverage determination, an authority approval, an automated gate – records its outcome as a value from a shared list (approved, denied, referred, partial approval…) and, where relevant, its reason (e.g. a denial reason). Each adjudication step applies business rules and produces a decision; the decision trace keeps the inputs, rules and cited sources that explain it.

Why we chose it. OMG DMN defines a decision as determining an output from inputs using decision logic; storing the outcome as a code (not a separate class per outcome) keeps reporting consistent across departments; the decision trace follows W3C PROV so an auditor can see why a decision was made.

What changed from today's CLM model. CLM had separate classes such as 'Claim Denial' and 'SIU Referral Decision'; they are now outcomes and reasons of one decision concept.

Please confirm

- Are the decision outcomes (Approved, Denied, Referred, Partial Approval…) complete for your department?
- Which decisions in your process must always carry a decision trace for audit?

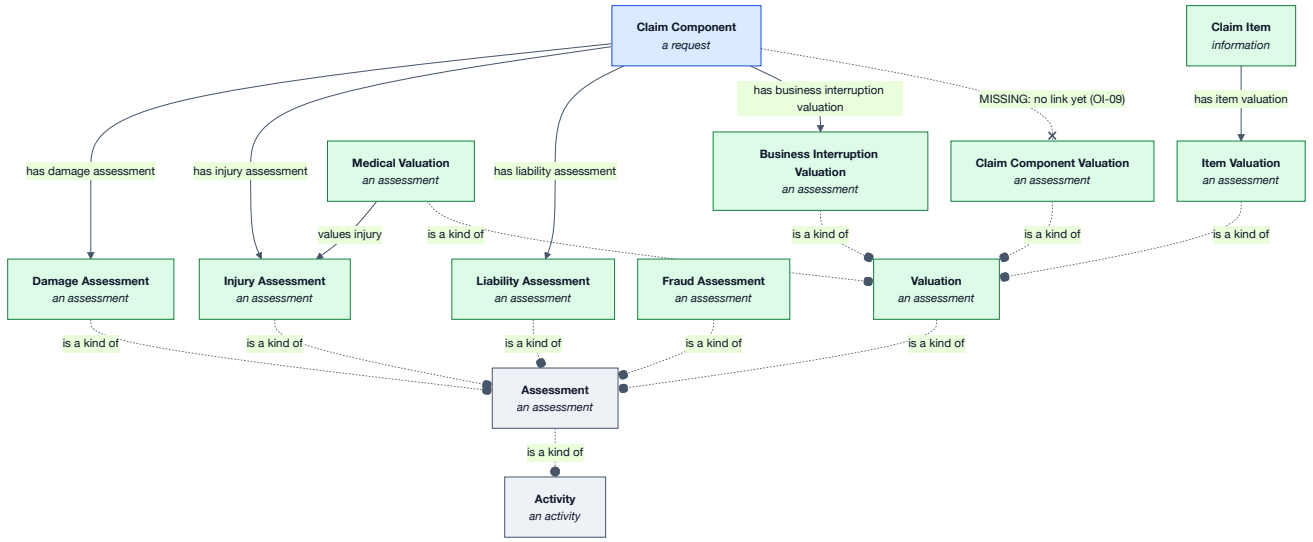
Your decision on A06 – Every decision records its outcome, its reason and how it was reached

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A07 – Assessments find the facts; valuations put a price on them



A07 – Assessments find the facts; valuations put a price on them

What this says. An assessment evaluates part of a claim – the damage, the injury, liability, fraud risk – and records findings. A valuation is the assessment that puts a monetary value on it: per claim component (gross loss, exclusions, limit and deductible applied, net payable), per item (replacement cost to actual cash value), for a bodily injury, or for business interruption.

Why we chose it. The OMG P&C Data Model and IBM IIW both treat assessments as their own concept, separate from the decision that uses them; a department adds its own kind of assessment or valuation underneath without changing the shared ones.

What changed from today's CLM model. Damage Evaluation became Damage Assessment; the four CLM valuation classes were placed under one Valuation. There is still no direct link from a claim component to its component valuation (shown in red).

Still to be fixed here

- **OI-09** Claim classes a query cannot reach from the claim: `clm:ComponentValuation`

Please confirm

- Which assessments and valuations does your department perform that are not listed?
- Should every claim component have exactly one component valuation?

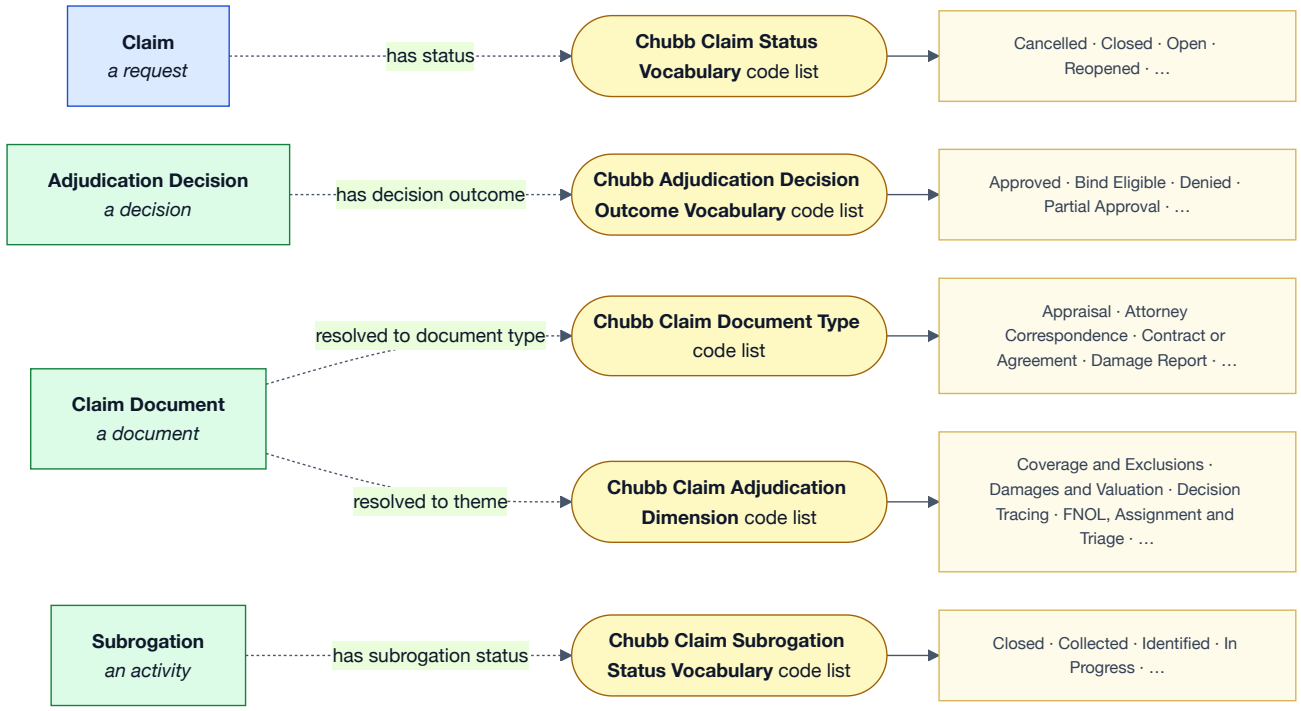
Your decision on A07 – Assessments find the facts; valuations put a price on them

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A08 – Codes come from shared, managed code lists



A08 – Codes come from shared, managed code lists

What this says. Codes such as a claim status, a decision outcome or a document type are not free text: each comes from a managed code list with a label, a definition and the source-system code. Enterprise lists (claim status, document type, payment type…) are shared by every department; a department can add its own list for codes that only it uses.

Why we chose it. W3C SKOS is the standard for code lists; W3C SHACL lets the platform reject a value that is not in the list. Using one list per kind of code is what makes a claim status mean the same thing in every report.

What changed from today's CLM model. 31 CLM vocabularies were promoted, merged or split into 69 enterprise code lists. 13 of them are still empty and 312 coded properties still need their list created – both are open items.

Still to be fixed here

- **OI-06** Relationships and attributes that say the same thing twice: `ins:hasStatus`

Please confirm

- Which code lists must be enterprise-wide, and which may each department own?
- Who owns each code list and approves new values?

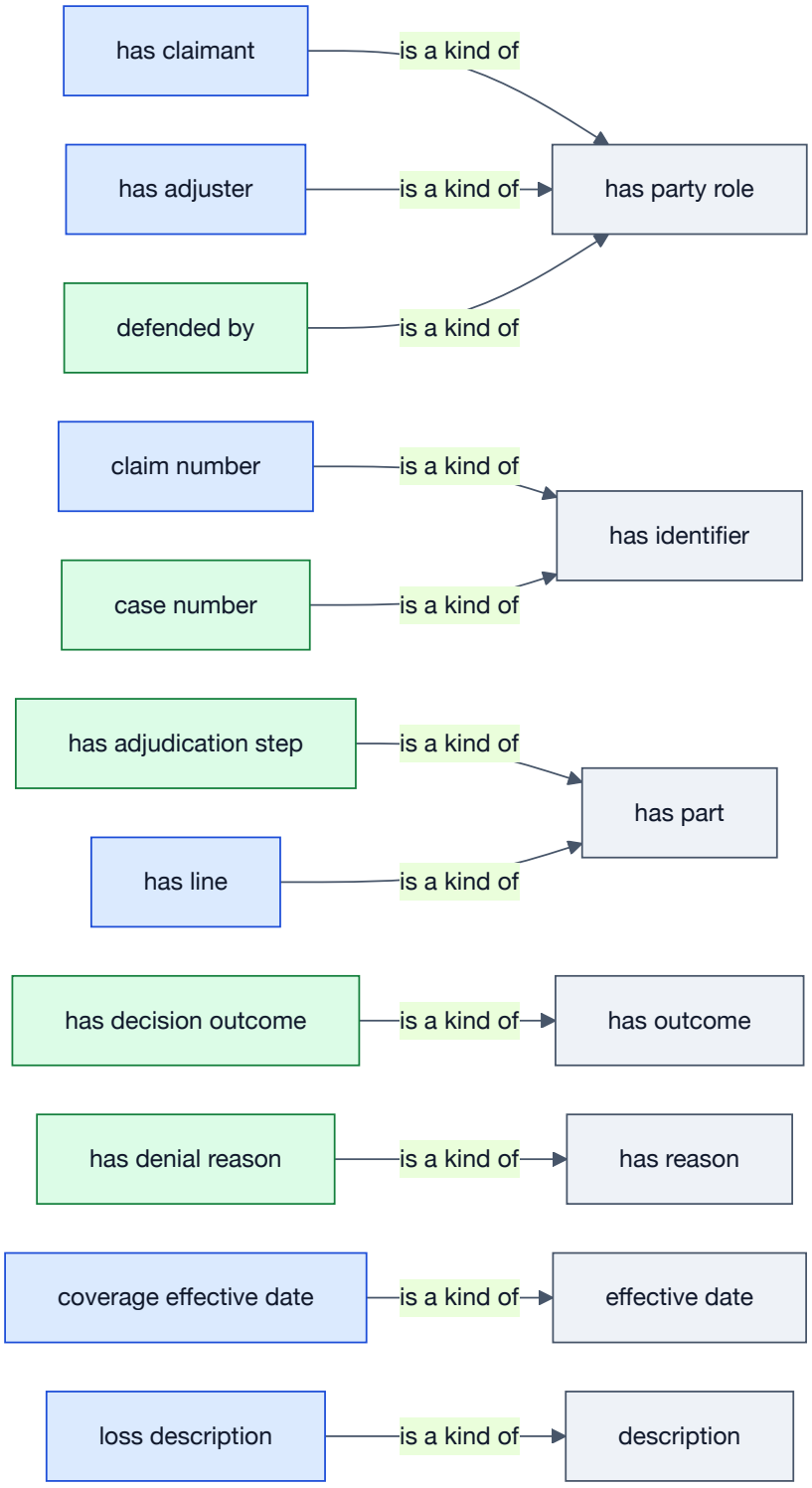
Your decision on A08 – Codes come from shared, managed code lists

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A09 – Shared kinds of link let every department ask the same question



A09 – Shared kinds of link let every department ask the same question

What this says. Specific links such as 'has claimant' or 'has adjuster' are declared as kinds of a general link ('has party role'). Specific numbers such as a claim number or a case number are kinds of 'identifier'. A report can therefore ask 'every party role on this claim' or 'every identifier of this record' once, for every department.

Why we chose it. FIBO defines properties at the highest level where they apply and specialises them below; the W3C RDF Schema sub-property rule makes the general question return the specific answers automatically.

What changed from today's CLM model. In CLM each link stood alone; 150 links are now grouped under nine shared ones.

Still to be fixed here

- **OI-06** Relationships and attributes that say the same thing twice: `ins:hasAdjuster`

Please confirm

- Are there links in your department that should be grouped the same way (e.g. other identifiers)?

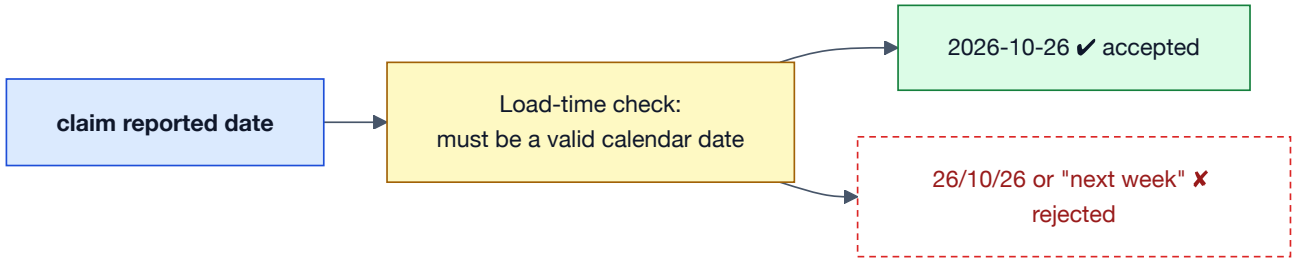
Your decision on A09 – Shared kinds of link let every department ask the same question

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A10 – How dates are checked



A10 – How dates are checked

What this says. Dates such as the date a claim was reported are stored as calendar dates and checked when data is loaded: a value that is not a valid date is rejected. Timestamps (date and time) are stored separately as date-times.

Why we chose it. The W3C OWL 2 standard does not include a plain 'date' type, so the reasoner sees the value as a literal while the W3C SHACL check enforces that it is a valid date. Nothing changes for the business user.

What changed from today's CLM model. Many CLM dates were stored as text; they are now proper dates or date-times. ins:dateOfLoss is still text and is in the open-items list.

> **Please confirm**
> - Which dates in your department are dates, and which must keep the time of day?

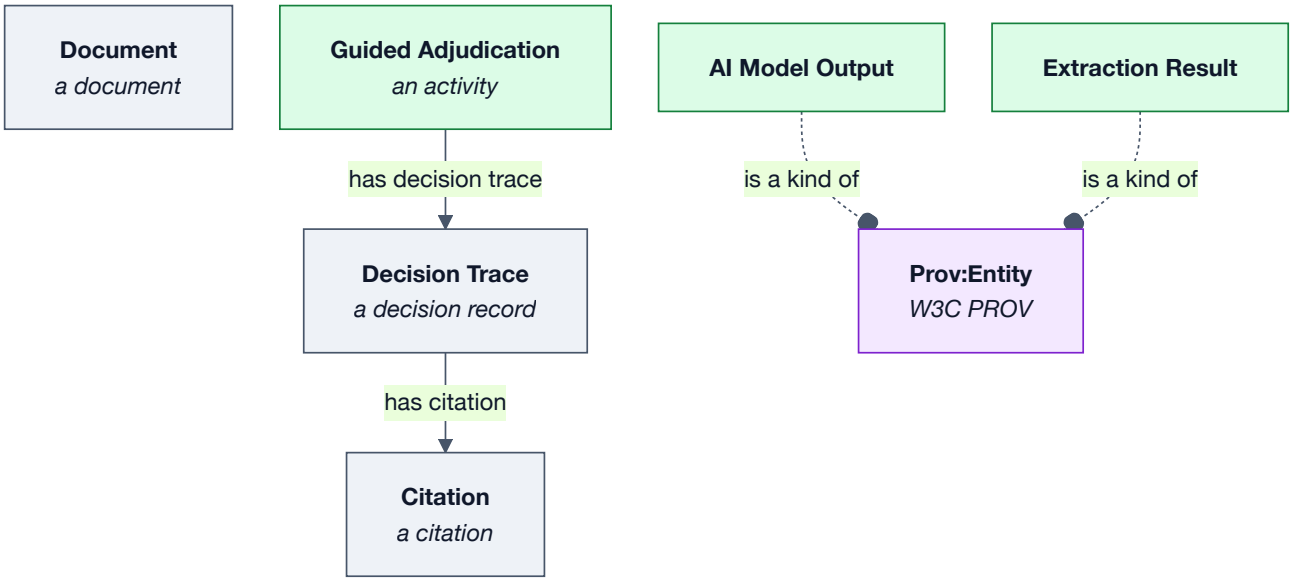
Your decision on A10 – How dates are checked

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

A11 – Every automated result says where it came from



A11 – Every automated result says where it came from

What this says. When a decision is made or a fact is extracted by AI, the system keeps where it came from: the decision trace lists the inputs and rules, and each citation points to the exact page or clause of the source document. Technical lineage (source table, ETL version) is kept in a separate provenance layer so the business model stays readable.

Why we chose it. W3C PROV-O is the standard for provenance; separating lineage from meaning is what lets several source systems feed the same enterprise concept.

What changed from today's CLM model. 245 lineage attributes (source ids, ETL versions, AI confidence values) moved out of the business ontology into the provenance layer.

Still to be fixed here

- **OI-08** Classes with no attributes or relationships of their own: `clm:ExtractionResult, clm:ModelOutput`
- **OI-13** Pipeline provenance classes still minted in the claims module: `clm:ExtractionResult, clm:ModelOutput`

Please confirm

- Which AI-extracted facts must be shown to an adjuster with their source citation?

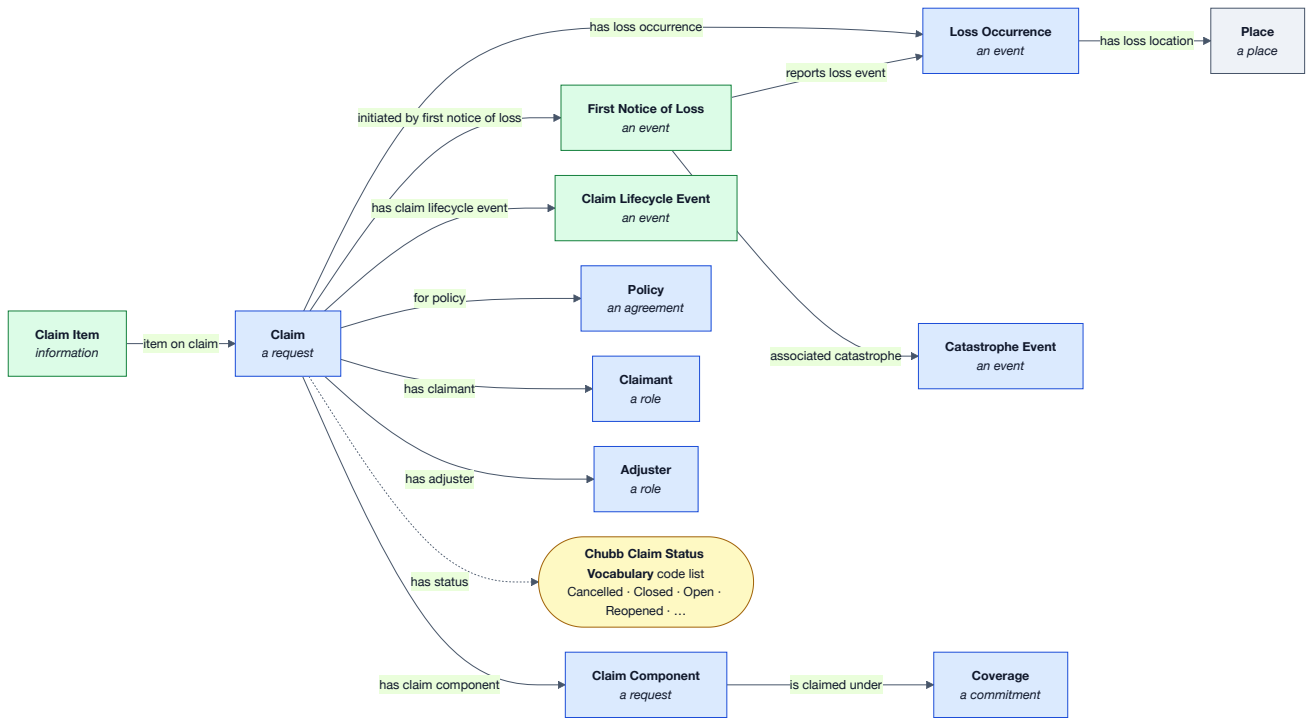
Your decision on A11 – Every automated result says where it came from

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C01 – The claim and what it is about



C01 – The claim and what it is about

What this says. A claim is made under a policy, reported by a first notice of loss, and caused by a loss occurrence that happened at a place (possibly part of a declared catastrophe). It is split into claim components, each claimed under one coverage. The claim has a claimant, an adjuster, a status from the claim-status list, lifecycle events (opened, closed, reopened) and the items involved.

Why we chose it. This is the enterprise claim kernel every department shares: ACORD, OMG P&C and IBM IIW use the same core (claim, claim component, occurrence, coverage, roles).

What changed from today's CLM model. Claim, Claim Exposure, Claim Line and Claim Feature from CLM are now Claim and Claim Component; the claim kernel moved from the claims module to the enterprise layer (ins:).

Still to be fixed here

- **OI-06** Relationships and attributes that say the same thing twice: `ins:forPolicy, ins:hasAdjuster, ins:hasStatus`
- **OI-08** Classes with no attributes or relationships of their own: `ins:Claimant`

Please confirm

- Is anything your department needs on every claim missing from this kernel?
- Is 'claim status' (Open, Closed, Reopened, Cancelled, Withdrawn) the full enterprise list?

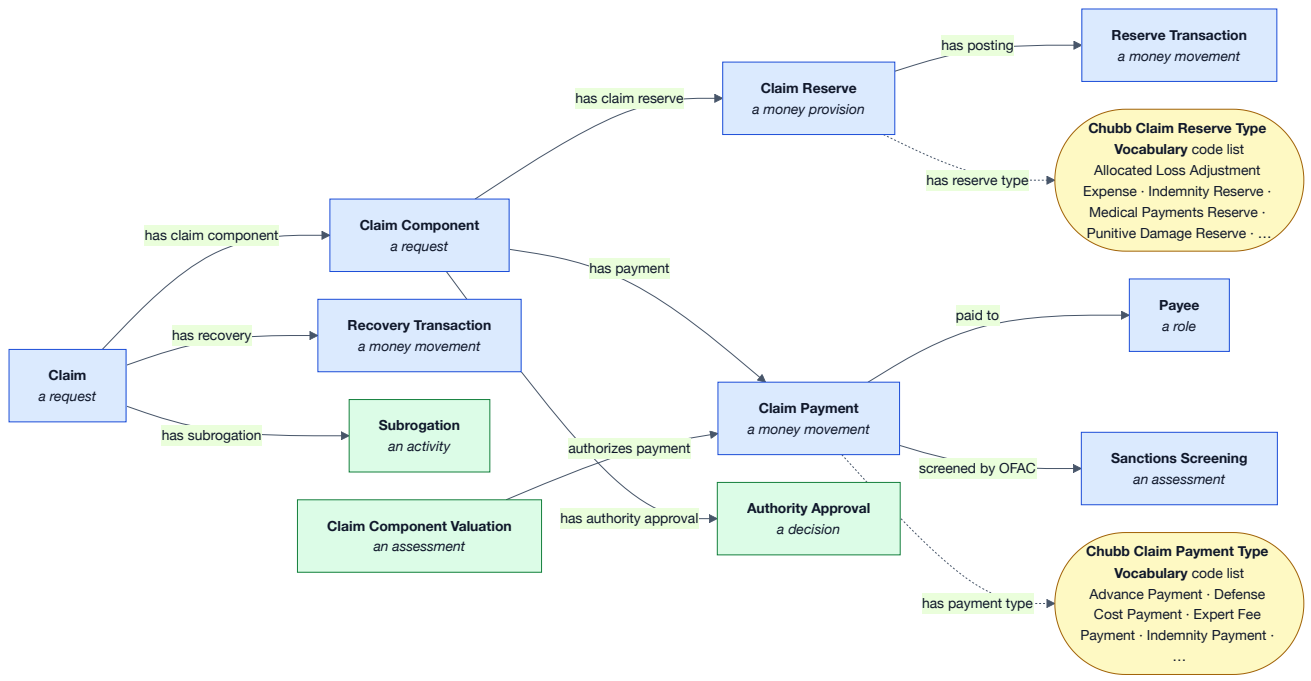
Your decision on C01 – The claim and what it is about

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C02 – Reserves, payments and recoveries



C02 – Reserves, payments and recoveries

What this says. Each claim component carries its reserves; every change to a reserve is a dated posting. Payments are made on a claim component to a payee, after a sanctions (OFAC) screening, and are authorised by the component valuation. Recoveries (subrogation, salvage) reduce the net cost of the claim; subrogation is the activity of pursuing a responsible third party. Authority approvals record when a handler needed approval above their authority.

Why we chose it. Reserves, payments and recoveries are enterprise financial concepts shared with Finance and Reinsurance, so they live in the enterprise layer (ins:); the reserve / payment / recovery types are enterprise code lists.

What changed from today's CLM model. Claim Reserve, Reserve Change, Claim Payment and Claim Recovery were promoted from CLM to the enterprise layer and re-based on the money pattern (A05).

Still to be fixed here

- **OI-06** Relationships and attributes that say the same thing twice: `ins:hasPayment`
- **OI-08** Classes with no attributes or relationships of their own: `ins:Payee`
- **OI-09** Claim classes a query cannot reach from the claim: `clm:ComponentValuation`
- **OI-16** Retired IRIs reused for a different meaning: `ins:hasPayment`

Please confirm

- Are reserves held per claim component in every department, or also per claim?
- Which payments need an authority approval in your department?

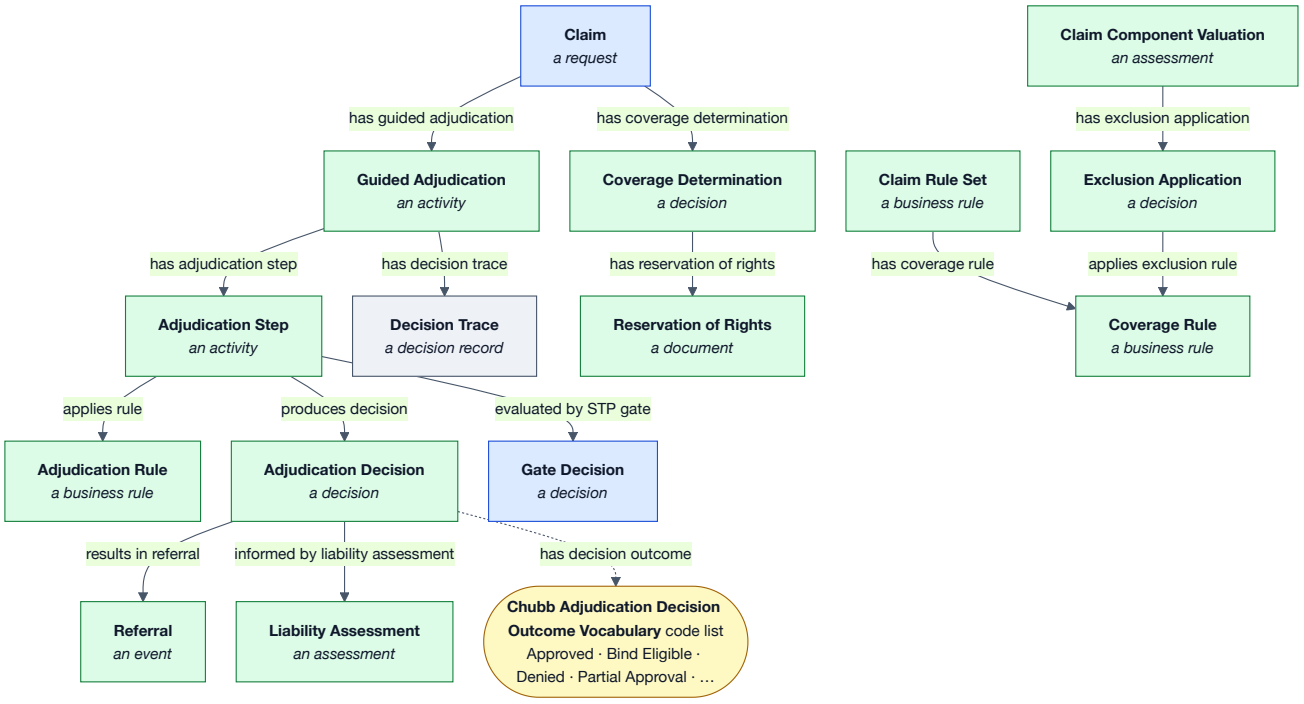
Your decision on C02 – Reserves, payments and recoveries

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C03 – How a claim is adjudicated



C03 – How a claim is adjudicated

What this says. Guided adjudication is the activity that takes a claim from first notice to closure in ordered steps. Each step applies adjudication rules, may pass through an automated straight-through-processing gate, and produces a decision with an outcome (approve, partially approve, refer, deny). A decision can lead to a referral and is informed by the liability assessment. Separately, the coverage determination decides whether the claim is covered, possibly under a reservation of rights. Coverage rules are grouped in rule sets per line of business and jurisdiction; exclusions are applied in the component valuation. The decision trace records how it was decided.

Why we chose it. This is the shared adjudication framework every claim department plugs into: a department adds its own rule sets, steps and decision outcomes as data, without changing the framework. It follows OMG DMN (decisions), BPMN / CMMN (steps and case handling) and ACORD Claim Activity.

What changed from today's CLM model. STP gate and STP score were merged into one gate decision; Claim Denial merged into the coverage determination (a denial is now a denial reason); SIU Referral merged into Referral.

Still to be fixed here

- **OI-09** Claim classes a query cannot reach from the claim: `clm:ComponentValuation`

Please confirm

- Can your department's adjudication be expressed as steps, rules and decisions in this framework?
- Should rule sets be keyed by line of business and jurisdiction only, or also by product?

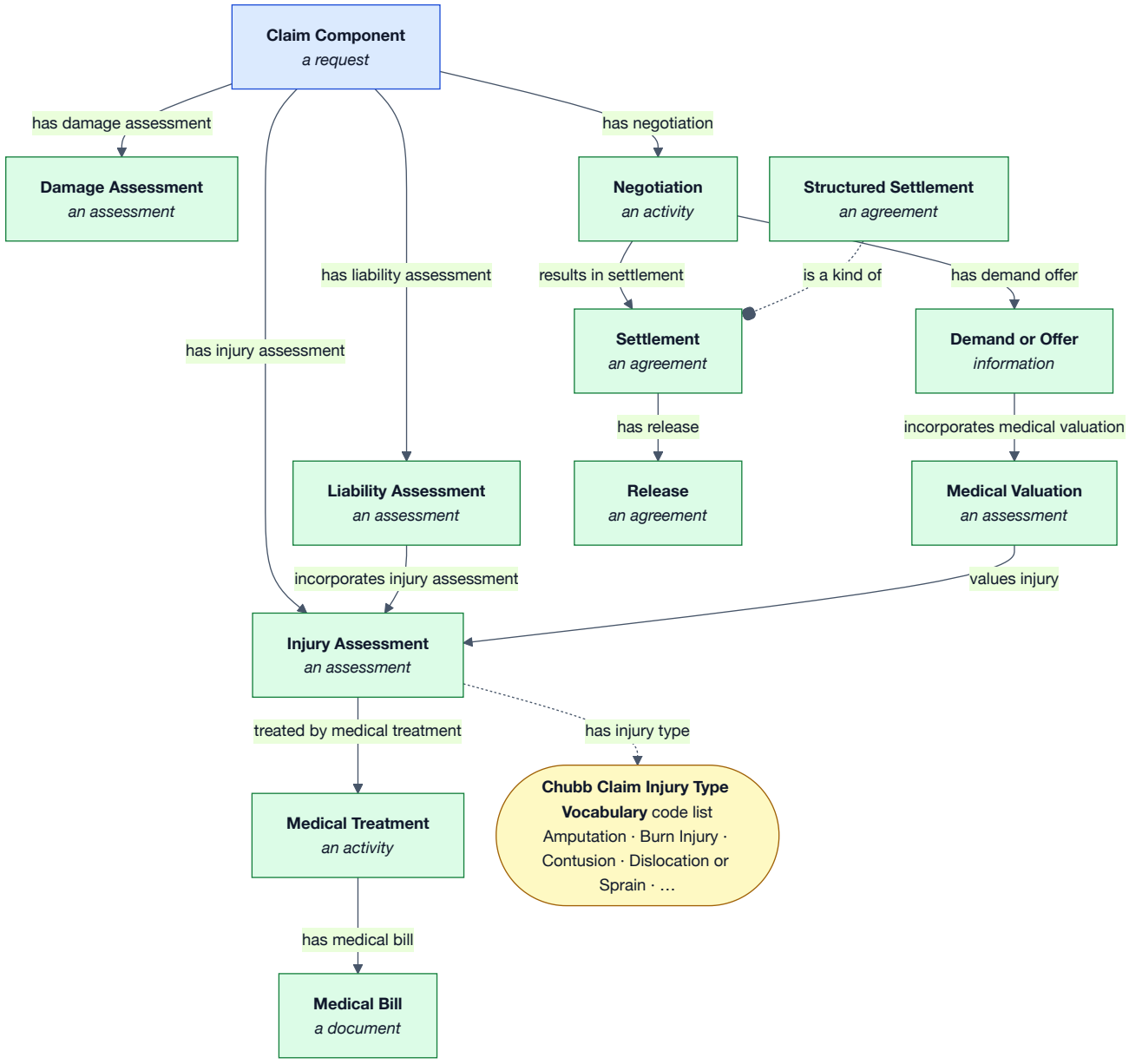
Your decision on C03 – How a claim is adjudicated

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C04 – Evaluating the loss and settling it



C04 – Evaluating the loss and settling it

What this says. Each claim component is assessed for damage, injury and liability; liability takes the injury into account. Injuries are valued (medical valuation) from the treatments and medical bills. Negotiation exchanges demands and offers – which can include the medical valuation – and ends in a settlement, possibly structured over time and with a signed release.

Why we chose it. These are shared claim-handling concepts (OMG P&C 'Claim Offer', 'Claim Assessment'); each department adds its own assessment details underneath.

What changed from today's CLM model. AI-extracted injury indicators (amputation, paraplegia, permanent disability…) were moved to the provenance layer; whether the business fact should stay is an open decision for you (OI-05).

Still to be fixed here

- **OI-01** Definitions that only repeat the label: `clm:treatedByMedicalTreatment`

Please confirm

- Should facts like 'amputation occurred' be business attributes, values of an injury-severity list, or provenance only?
- Is a settlement always the end of one negotiation on one claim component?

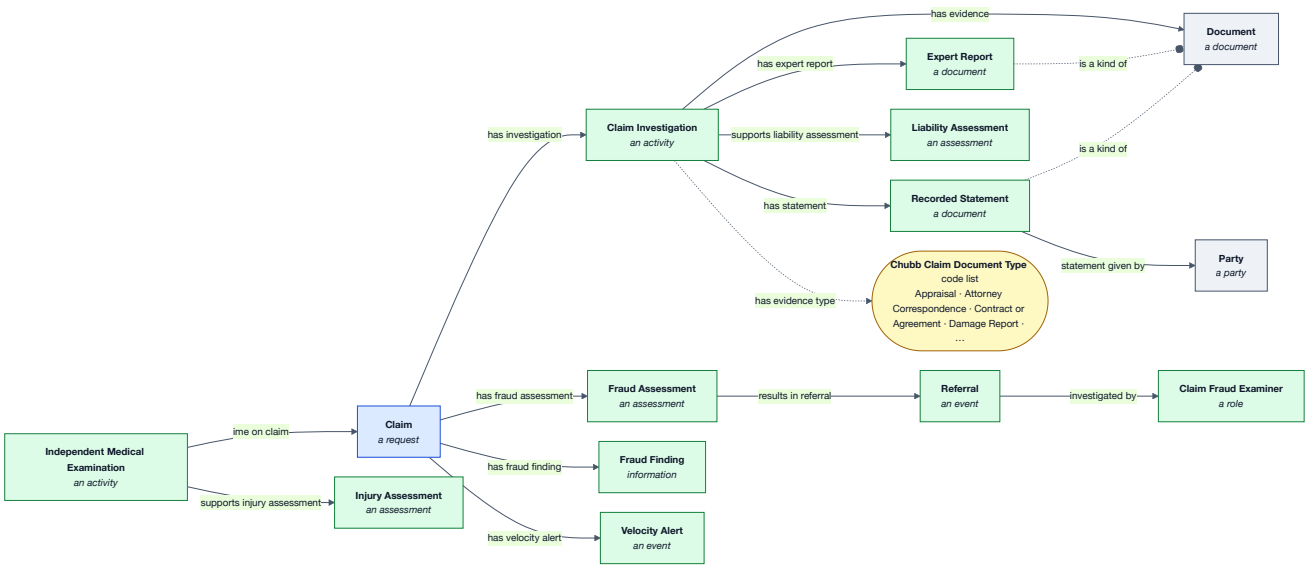
Your decision on C04 – Evaluating the loss and settling it

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C05 – Investigating the claim and handling suspected fraud



C05 – Investigating the claim and handling suspected fraud

What this says. An investigation establishes how the loss happened: it gathers evidence (documents or physical objects), recorded statements and expert reports, and supports the liability assessment. A fraud assessment can result in a referral to the SIU, investigated by a fraud examiner; fraud findings and velocity alerts are recorded on the claim. An independent medical examination supports the injury assessment.

Why we chose it. 'Being evidence' is a role a document or object plays in an investigation, not a kind of thing (OntoClean), so the investigation links directly to the document or object.

What changed from today's CLM model. Evidence is no longer a class; SIU Referral merged into Referral with a referral type; the evidence link is now called 'has evidence' (corrected in this build).

Still to be fixed here

- **OI-01** Definitions that only repeat the label: `clm:hasEvidence, clm:supportsLiabilityAssessment`
- **OI-08** Classes with no attributes or relationships of their own: `clm:ClaimFraudExaminer`
- **OI-10** Facts the business questions ask for that the model cannot hold: `clm:VelocityAlert`

Please confirm

- Is a referral always triggered by a decision or a fraud assessment, or can an adjuster refer directly?
- Which velocity-alert thresholds does your department use (the model cannot hold the threshold yet – OI-10)?

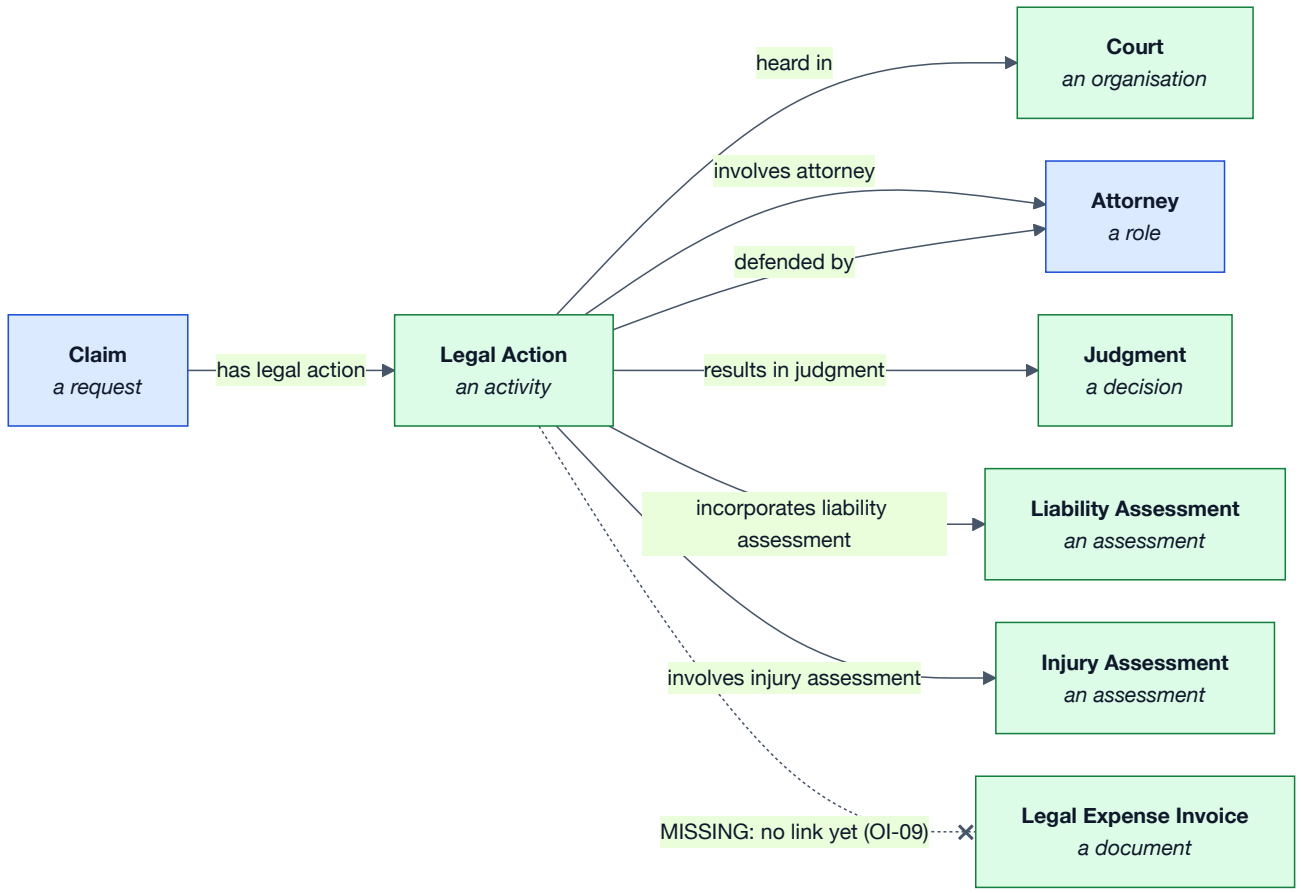
Your decision on C05 – Investigating the claim and handling suspected fraud

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C06 – Legal actions on a claim



C06 – Legal actions on a claim

What this says. A legal action (lawsuit, arbitration, mediation or appraisal) arises from a claim, is heard in a court, involves attorneys (one of whom defends), takes the liability and injury assessments into account and may end in a judgment. Legal expense invoices record what the defence costs.

Why we chose it. IBM IIW's 'Legal Action' covers every dispute-resolution proceeding, so one concept serves lawsuits, arbitrations and mediations, told apart by the proceeding type.

What changed from today's CLM model. Claim Litigation, Lawsuit and Legal Proceeding were three overlapping CLM classes; they are now one Legal Action. Legal expense invoices are not yet linked to the legal action (shown in red).

Still to be fixed here

- **OI-09** Claim classes a query cannot reach from the claim: `clm:LegalExpenseInvoice`
- **OI-10** Facts the business questions ask for that the model cannot hold: `clm:Judgment`

Please confirm

- Do arbitration and mediation need anything a lawsuit does not?
- Should the judgment date and the defence cost per legal action be recorded (currently missing)?

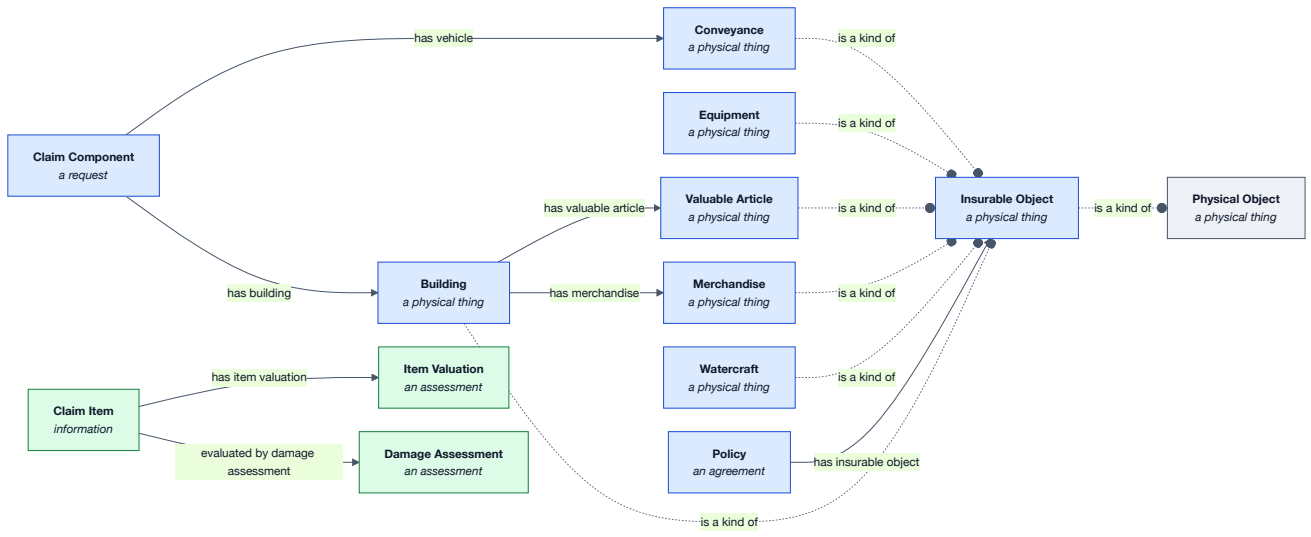
Your decision on C06 – Legal actions on a claim

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C07 – The things that were damaged



C07 – The things that were damaged

What this says. Insurable objects – buildings, vehicles, equipment, merchandise, valuable articles, watercraft – are physical things insured by a policy and involved in claim components. A claim item records the involvement of an object in a claim with its alleged damage, its damage assessment and its valuation.

Why we chose it. The OMG P&C model and IBM IIW keep the insured object separate from its involvement in a claim; a department adds its own kinds of object underneath (e.g. aircraft, crops).

What changed from today's CLM model. Claim Property was a duplicate of Building and was merged; Workers' Compensation Item was split into a claim component, equipment and a body-part code.

Please confirm

- Which insured objects does your department need that are not listed?

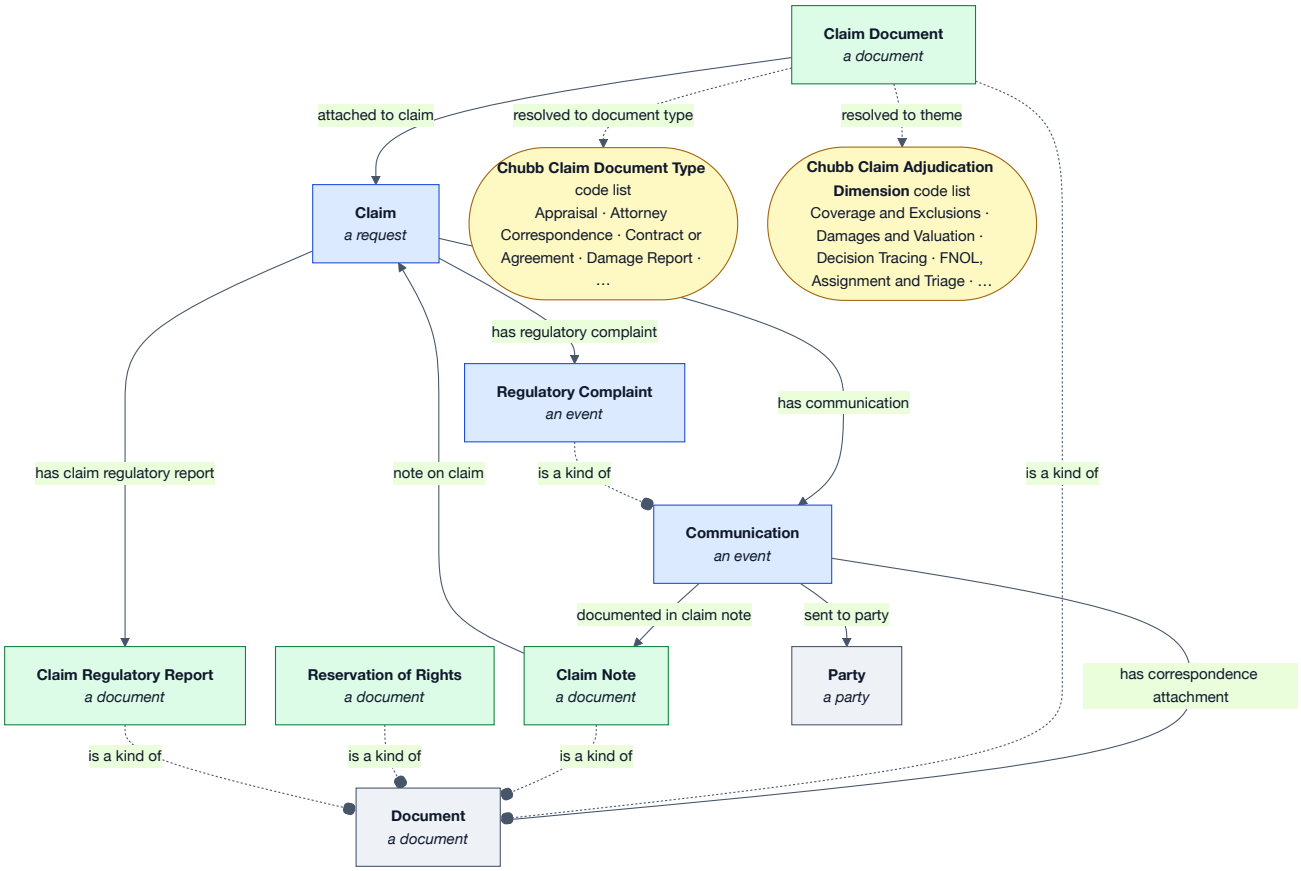
Your decision on C07 – The things that were damaged

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C08 – Documents, notes and correspondence



C08 – Documents, notes and correspondence

What this says. Documents held in the claim file are typed from the enterprise document-type list and tagged with the adjudication dimension they support. Notes record what the handler observed or did; communications (letters, e-mails, calls) are sent to parties, may carry attachments and are documented in notes. Regulatory reports and Department of Insurance complaints are recorded on the claim.

Why we chose it. OMG Commons defines Document, Record and Notice; a single document concept with a type code keeps every department's documents searchable in one way.

What changed from today's CLM model. Claim Attachment became Claim Document; document-routing and extraction profiles moved to the application configuration graph.

Still to be fixed here

- **OI-01** Definitions that only repeat the label: `clm:documentedInClaimNote, ins:sentToParty`
- **OI-10** Facts the business questions ask for that the model cannot hold: `clm:ClaimRegulatoryReport, ins:RegulatoryComplaint`

Please confirm

- Is the document-type list complete for your department?

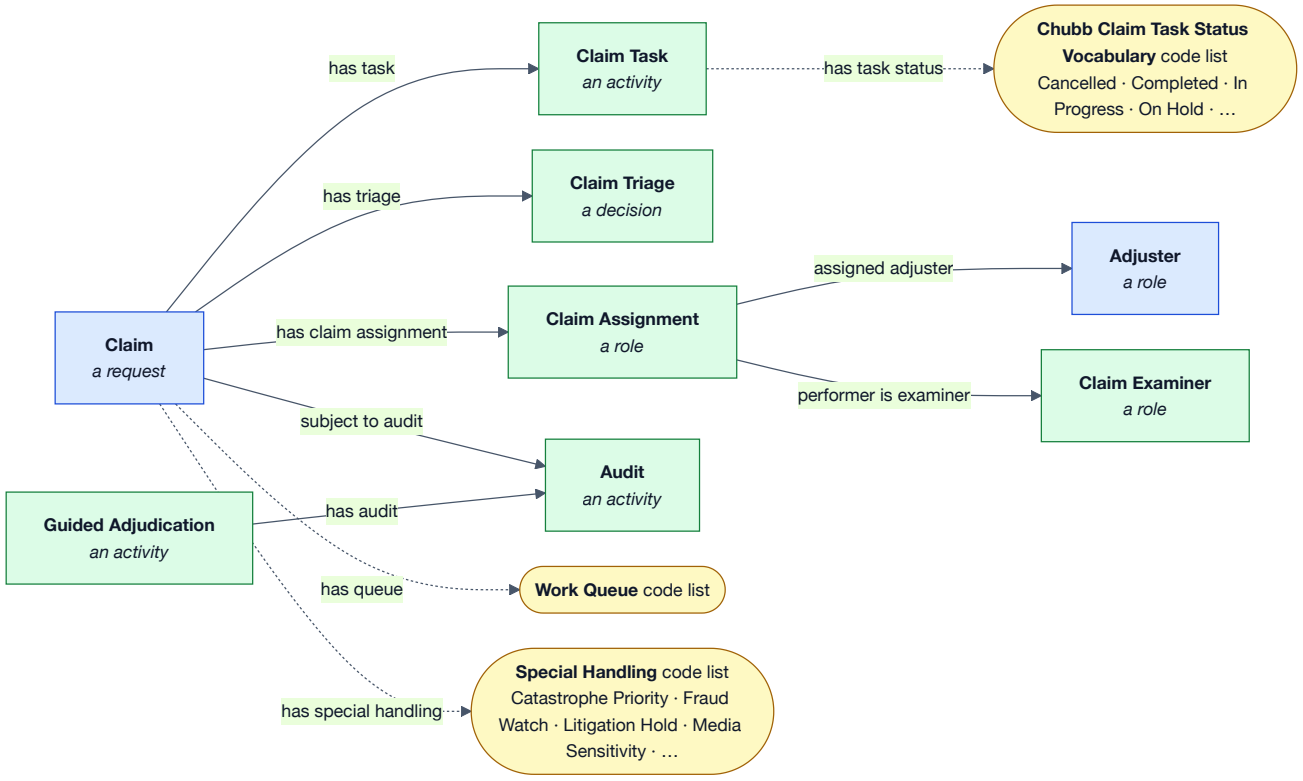
Your decision on C08 – Documents, notes and correspondence

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C09 – Who handles the claim and what they must do



C09 – Who handles the claim and what they must do

What this says. Triage routes a claim to a handling segment and queue; assignments record which adjuster or examiner handles the claim in which capacity; tasks (diary items) are what the handler must do by a due date; audits review the file against handling standards. Special-handling flags (catastrophe priority, litigation hold…) come from a shared list.

Why we chose it. These handling concepts are common to every claim department; work queues and special-handling values are where departments differ, so they are code lists a department can extend.

What changed from today's CLM model. 'Has diary' and 'has task' still both exist and mean the same thing (open item OI-06).

Still to be fixed here

- **OI-04** Code lists created empty (skeleton) – values still to be loaded: `vocabulary/clm/work-queue/`
- **OI-06** Relationships and attributes that say the same thing twice: `clm:hasTask`
- **OI-08** Classes with no attributes or relationships of their own: `clm:ClaimExaminer`

Please confirm

- Which work queues and special-handling values does your department need?

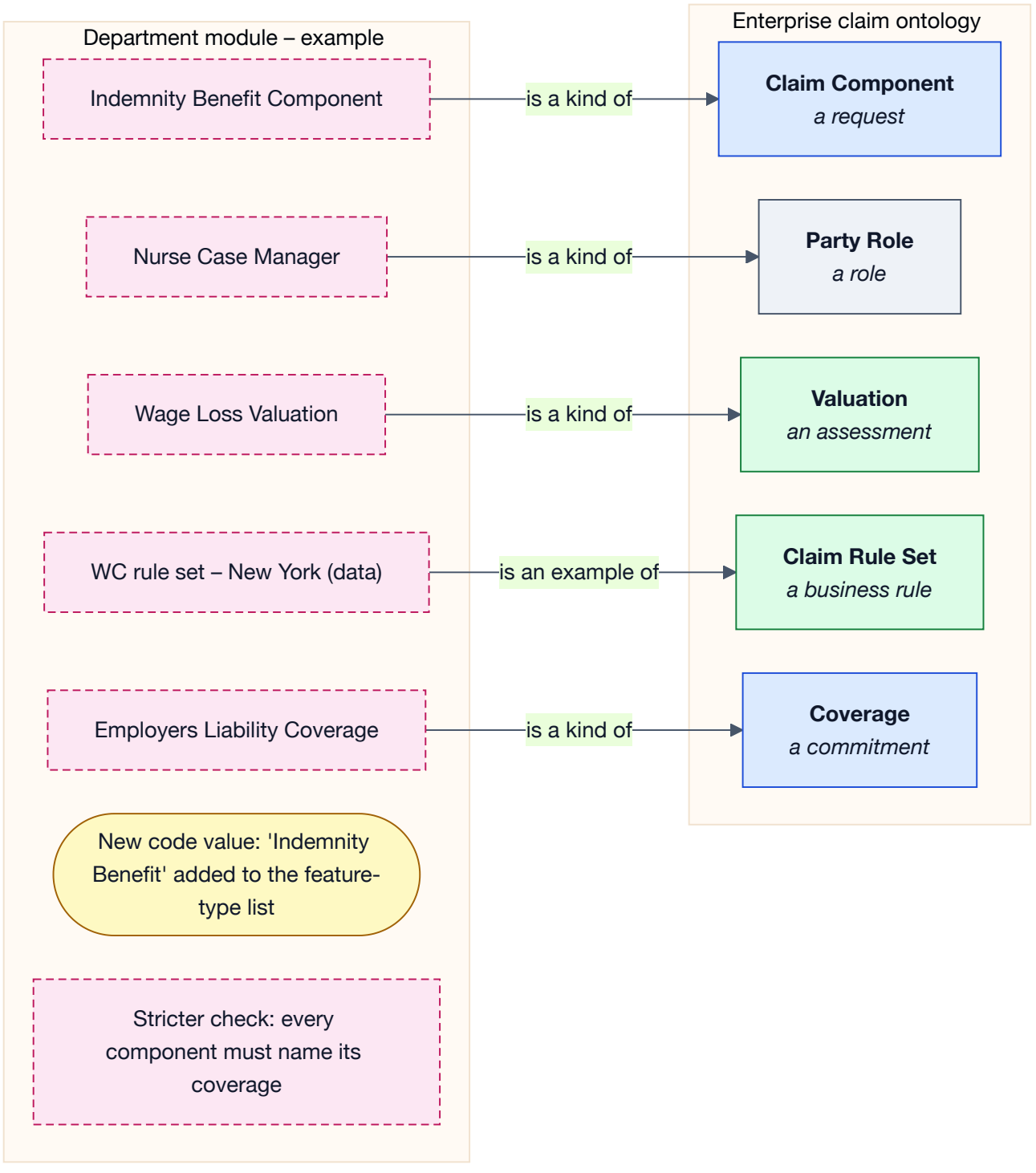
Your decision on C09 – Who handles the claim and what they must do

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

C10 – How your department adds its own terms (example)



C10 – How your department adds its own terms (example)

What this says. A department never changes an enterprise term. It creates its own module (for example 'Workers' Compensation claims') that imports the enterprise ontology and adds, underneath it: kinds of claim component that need their own attributes, department roles, department valuations, its rule sets (as data),

department code values, and stricter validation shapes. Everything a department adds is still found by an enterprise query, because it sits under an enterprise class.

Why we chose it. This is how FIBO and the OMG specifications are extended by users: import, subclass, specialise properties, never edit the imported files. SHACL shapes in the department module may tighten the rules but not relax them.

What changed from today's CLM model. New – the CLM ontology had no extension mechanism; departments would have copied it.

Please confirm

- Which department will pilot an extension module first?
- Who in each department approves its extension before it is published?

Your decision on C10 – How your department adds its own terms (example)

☐ I agree ☐ I agree with the changes noted below ☐ I do not agree ☐ Not my area

Changes / comments:

Name | Department / line of business | Date

4. Decisions only you can make

These open items need a business answer before the release. The full list of affected terms for each is in the workbook `workbook/Chubb_Claim_Enterprise_Ontology_Status.xlsx`, sheet ‘Open Item Terms’ (filter by the item id).

OI-01 · Definitions that only repeat the label P2 · 30 item(s) · owner: SME

An ISO 704 definition (genus + differentia) is what lets every department read the term the same way; 'Demand Date.' says nothing.

What we propose: SME writes a one-sentence definition; the engineer stores it as skos:definition and removes the editorial note.

`clm:addressesInjuryAssessment, clm:auditDate, clm:auditQuestion, clm:demandDate, clm:documentedInClaimNote, clm:hasEvidence, clm:offerDate, clm:supportsLiabilityAssessment, clm:treatedByMedicalTreatment, clm:vendorReport …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-02 · Definitions generated from the label (source held only lineage) P2 · 22 item(s) · owner: SME

The source definition was only system lineage, so the build wrote a placeholder from the label.

What we propose: SME confirms or rewrites the generated sentence.

`clm:fileNoteTextLastUpdateTimestamp, clm:fileNoteTextWithHtmlTags, clm:hasAuthorityLimitType, clm:hasAuthorityStatus, clm:hasInjuryStatus, clm:hasLiabilityPercent, clm:hasLitigationType, clm:noteAttentionRequiredIndicator, clm:noteUpdatedTimestamp, clm:numberOfTreatments …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-03 · Code values whose definitions were generated from the label P3 · 72 item(s) · owner: SME

Code values are shared across departments; a value without a definition is used inconsistently.

What we propose: SME confirms each value's meaning and gives the source-system code (skos:notation).

`vocabulary/clm/claim-lifecycle-event-type/Closed, vocabulary/clm/claim-lifecycle-event-type/Opened, vocabulary/clm/claim-lifecycle-event-type/Reopened, vocabulary/clm/claim-regulatory-report-type/CMSSection111Report, vocabulary/clm/claim-regulatory-report-type/ISOClaimSearchReport, vocabulary/clm/claim-regulatory-report-type/StatutoryClaimReport, vocabulary/clm/fraud-finding-type/DocumentFraudFlag, vocabulary/clm/fraud-finding-type/FraudNetworkRing, vocabulary/clm/legal-action-type/Lawsuit, vocabulary/clm/referral-type/Escalation …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-04 · Code lists created empty (skeleton) – values still to be loaded P1 · 13 item(s) · owner: SME + data owner

A property constrained to an empty code list cannot be populated; SHACL rejects every value.

What we propose: Load the values (with skos:notation and definitions) from the source system or the ACORD code list; SME confirms.

`vocabulary/clm/advocacy/, vocabulary/clm/body-part/, vocabulary/clm/work-queue/, vocabulary/fnd/legal-entity-type/, vocabulary/ins/construction-type/, vocabulary/ins/industry-vertical/, vocabulary/ins/item-detail-type/, vocabulary/ins/market-segment/, vocabulary/ins/market-type/, vocabulary/ins/occupancy/ …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-05 · AI-extracted injury, liability and handling indicators moved to provenance P1 · 43 item(s) · owner: SME

These indicators (e.g. amputation, paraplegia, permanent disability) were defined as 'AI indicator extracted from adjuster notes', so the build moved them to the PROV layer. The business fact itself (an amputation occurred) now has no home in the ontology, while the adjuster's reasoning text was kept.

What we propose: Decide per indicator: (a) keep the fact as a business attribute and record the AI origin as PROV-O provenance, or (b) express it as a value of an injury-severity / handling-indicator code list (several already exist in vocabulary/clm/injury-type/).

`clm:amputationFlag, clm:catastrophicInjuryToChildFlag, clm:damagesWithinPrimaryLayerFlag, clm:distractedDrivingFlag, clm:drugInvolvementFlag, clm:drunkDrivingFlag, clm:economicDamageFlag, clm:futureWageLossFlag, clm:gangInvolvementFlag, clm:highAttachmentPointFlag …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-06 · Relationships and attributes that say the same thing twice P2 · 28 item(s) · owner: SME + ontology engineer

Two names for one fact split the data: one department fills one, another department the other.

What we propose: SME confirms each pair is one fact; the engineer keeps the survivor and maps the other to it (merge + crosswalk).

`clm:assignmentOnClaim, clm:depreciationAmount, clm:grossSettlementAmount, clm:hasDiary, clm:hasTask, clm:itemDepreciationAmount, clm:netPayable, clm:netPayableAmount, clm:performsOnClaim, clm:settlementAmount …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-08 · Classes with no attributes or relationships of their own P3 · 36 item(s) · owner: SME

A class with nothing to say may be a pure role marker (fine, e.g. ins:Claimant) or a concept whose attributes were lost in the merge.

What we propose: SME confirms each is intended as a marker, or names the attributes it needs.

`clm:AssignmentOfBenefits, clm:ClaimExaminer, clm:ClaimFraudExaminer, clm:ClaimPaymentAuthorization, clm:ExtractionResult, clm:ModelOutput, clm:RepairService, fnd:Identifier, ins:Account, ins:AdditionalInsured …`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-09 · Claim classes a query cannot reach from the claim P1 · 4 item(s) · owner: SME + ontology engineer

No chain of claim relationships (in either direction, not counting shared hubs such as amounts and code lists) connects these classes to ins:Claim; clm:ComponentValuation is reached only through a payment, never from the claim component it values – the competency question on exclusions had to leave that join open.

What we propose: Add the missing link (e.g. ins:ClaimComponent —clm:hasComponentValuation→ clm:ComponentValuation; clm:LegalAction —clm:hasLegalExpenseInvoice→ clm:LegalExpenseInvoice) or confirm the class is reached another way.

`clm:AssignmentOfBenefits, clm:ClaimPaymentAuthorization, clm:ComponentValuation, clm:LegalExpenseInvoice`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-10 · Facts the business questions ask for that the model cannot hold P2 · 7 item(s) · owner: SME + ontology engineer

Found while porting the competency questions: the question is valid but the class has no attribute for the fact.

What we propose: SME confirms the fact is needed; the engineer adds the attribute (or a code list) to the enterprise class.

`clm:ClaimRegulatoryReport, clm:Judgment, clm:VelocityAlert, ins:RegulatoryComplaint, ins:ReinsuranceCession, ins:Subjectivity, uw:Submission`

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-12 · Limits and deductibles held twice: as flat amounts and as coverage terms P2 · 13 item(s) · owner: SME + ontology engineer

The target pattern is Coverage —hasLimit→ Limit (amount + limit type); these flat amounts on Coverage / Policy say the same thing a second way.

What we propose: SME confirms the coverage-term pattern; the engineer turns each flat amount into a Limit / Deductible / SIR with its limit-type value.

```
ins:aggregateCoverageLimitAmount, ins:aggregateLimit, ins:eachOccurrenceLimitAmount,
ins:fireLegalLiabilityLimit, ins:medicalPaymentsLimit, ins:occurrenceLimit,
ins:perLocationAggregateLimit, ins:perProjectAggregateLimit, ins:personalAdvInjuryLimit,
ins:policyLevelDeductibleAmount …
```

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-17 · Codes still named after source columns P2 · 4 item(s) · owner: SME + ontology engineer

OMG P&C and FIBO naming rules require spelled-out business names; 'Claim A / B / C' only make sense to someone who knows the source table.

What we propose: Rename to the business name suggested (crosswalk entry kept).

```
ins:hasCat, ins:hasClaimA, ins:hasClaimB, ins:hasClaimC
```

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

OI-18 · Coded properties with no code list – the plan's 'new code lists' work package P2 · 308 item(s) · owner: Ontology engineer + SME

These properties were converted from text to codes, but their code lists were never created, so the SHACL shape only checks 'is a skos:Concept' and accepts a value from any list (a litigation type could be a claim status). This is the 'new code lists' work package of the two-sprint plan, still to do.

What we propose: Create each code list from the source-system values (ACORD code lists where they exist) and bind the property to it (sh:node … skos:inScheme). Where a list with the same name already exists it is named here – bind it now.

```
clm:hasAdjudicationStatus, clm:hasAuditOutcome, clm:hasAuthorityLimitType,
clm:hasAuthorityStatus, clm:hasAutomationLevel, clm:hasCPT, clm:hasChubbDocumentType,
clm:hasChubbFolderType, clm:hasCircumstance, clm:hasClosureReason …
```

☐ Agree with the proposal ☐ Agree with changes ☐ Disagree ☐ Not my area

Name | Department | Date

5. Overall sign-off

By signing, I confirm that I reviewed the pages marked above and that, apart from the changes and disagreements I noted, the enterprise claim ontology describes how my department's claims work, and that my department can build its claim adjudication by extending it as shown on page C10.

☐ Approved ☐ Approved with the changes noted ☐ Not approved – see comments

Overall comments:

Name | Department / line of business | Date

Signature | Role