CycloneDX

CycloneDX 2.0

From Bill of Materials to Transparency Exchange Language

OWASP

OWASP FOUNDATION

®

The question has changed

2018 — *What is in it?*

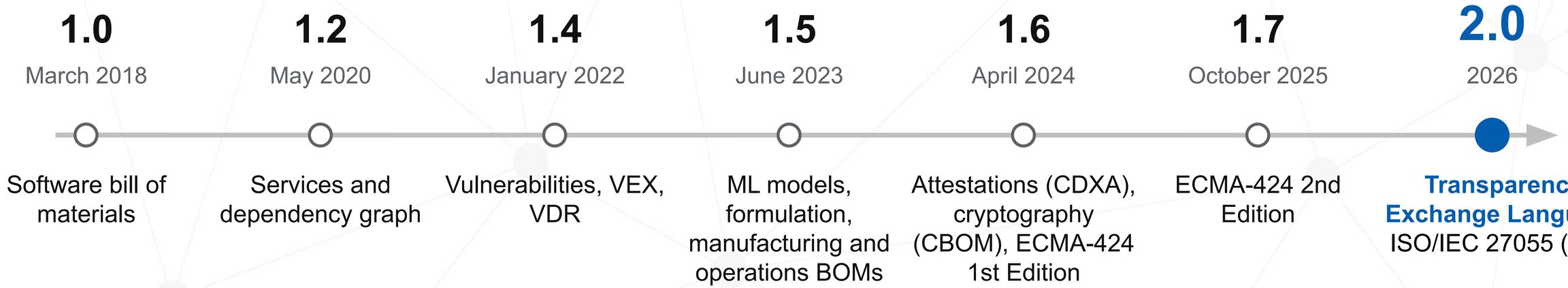
2020 — *Is it vulnerable?*

2023 — *Can you prove it?*

2026 — **What does it do, what could go wrong, who is accountable, and how do you know?**

Each question is a superset of the one before.

Eight years of evolution



Every release widened what CycloneDX could describe. 2.0 is the first release that changes what CycloneDX is.

Bill of Materials to Transparency Exchange Language

A BOM answers **“what.”** A transparency language answers **“what, how, why, and so what.”**

New first class citizens

`inventories` `threats` `risks` `controls` `blueprints` `profiles` `perspectives`

Act 1

Describe anything

what it is and what it does

→

Act 2

Prove it

substantiate every claim

→

Act 3

The transparency fabric

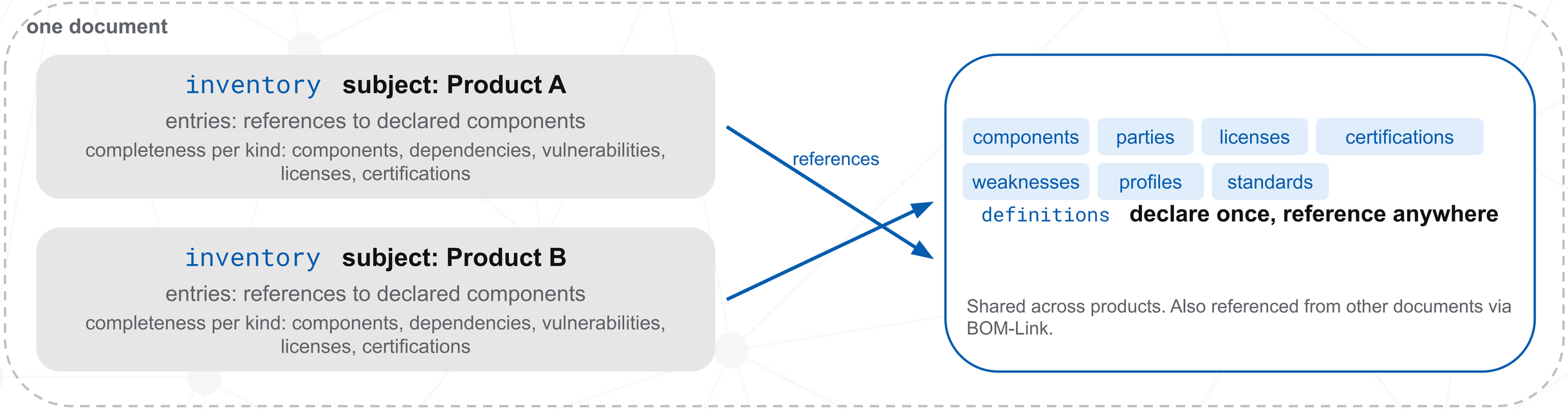
one graph, every team, every organization

Everything from 1.x carries forward: dependencies, compositions, vulnerabilities, formulation, declarations, citations, annotations.

Act 1: Describe anything

What it is and what it does

Inventories: one document, many subjects



“Complete, with no licenses” is a statement, not a silence.

Completeness is explicit per kind. Services are components. A party model replaces authors, suppliers, and manufacturers.

Hardware inventory

`device` **physical properties**

`deviceType` SMD, BGA, QFN, die, module
`boardLocation` U5, R12, C3
`classification` HS, ECCN, UNSPSC, NAICS, JEDEC
`leadTime` `quantity`

Types: device and material. Firmware remains its own type.

`componentChoice` **approved alternates, multisourced parts**

Certifications, hardware and software alike

CE | FCC Part 15 | UKCA | FIPS 140-3 | Common Criteria

Identifiers asserted per party

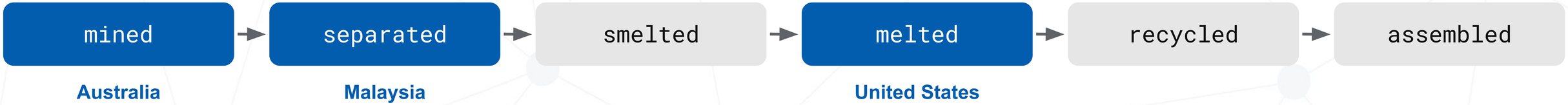
MPN | part number | GTIN | GLN | serial number | UDI | IMEI | MAC address | FCC ID | EPC RFID | and more

Down to the raw materials

`material` **component type** metals · alloys · polymers · chemicals

`materialForm` powder · ingot · wire · sheet · foil · sponge · and more

Stage scoped origins each stage is a distribution across countries, with quantities and units



`performedBy` names the party, for example the smelter. Example: a samarium cobalt magnet whose samarium was mined, separated, and melted in three countries.

Hazards are declared as weaknesses using GHS hazard statements, for example lead in solder.

Cryptography, from inventory to usage

Inventory
CBOM, since 1.6

algorithms

certificates

protocols

keys

related material

Usage, lifecycle, and agility new in 2.0

What it does

keygen, encapsulate, sign, verify, key agree, wrap. Security properties: IND-CCA2, EUF-CMA, collision resistance.

Lifecycle

Key states per NIST SP 800-57. Rotation and renewal with their mechanisms.

As configured or observed

Protocols and cipher suites recorded as configured or observed, not just as supported.

Certifications such as FIPS 140-3 are recorded the same way for crypto, hardware, and software.

AI as a first class subject

Model

Properties
learning types, tasks, architecture, parameter count, quantization, inputs and outputs, training, limitations

Measurements

On any component
performance metrics with slices and confidence intervals, energy in kWh, emissions in tCO2eq

Units
UCUM, UNECE Recommendation 20, QUDT

Agents

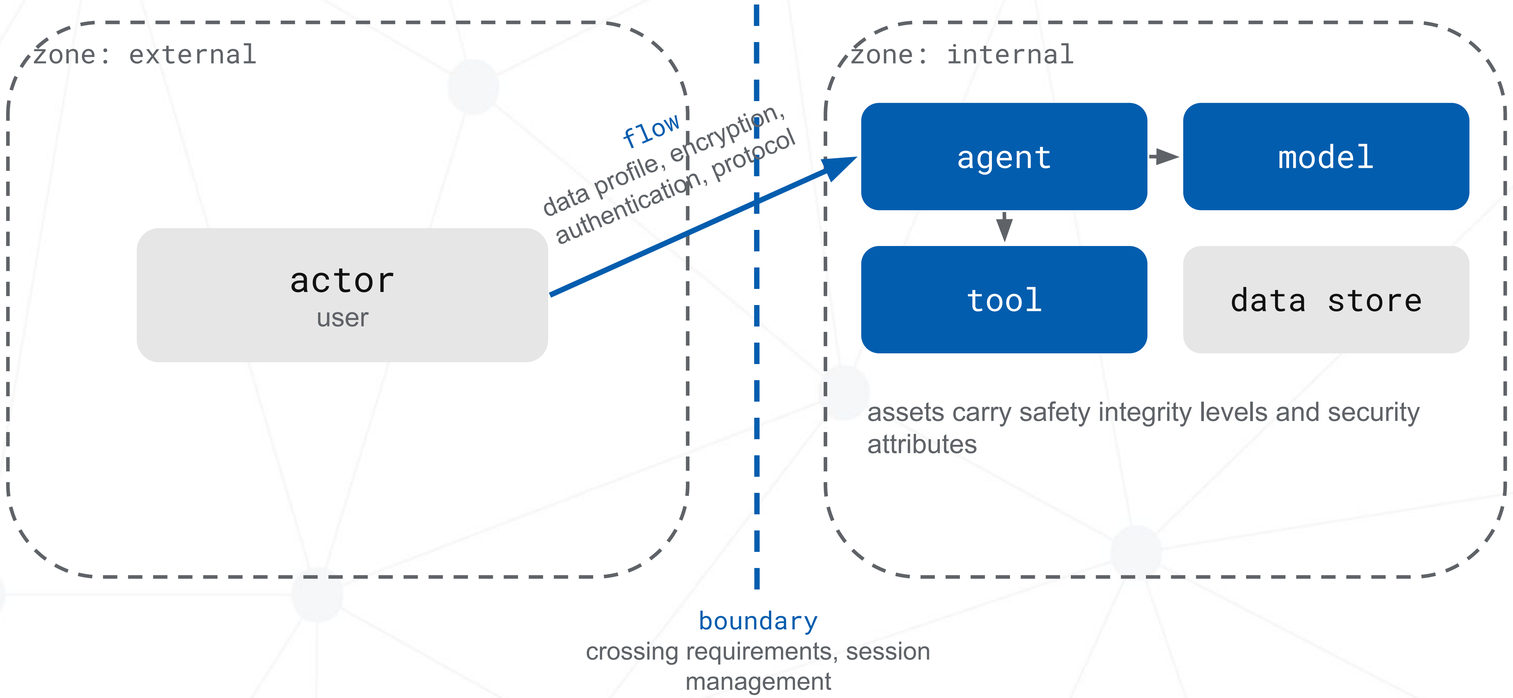
Parties
with delegation from the human they act for

Behaviors
plans action, delegates task, invokes tool, requests user approval, coordinates agents

Blueprint assets
agent, model, tool

Ethics, fairness, and groups at risk become **risks**. The model card becomes a **perspective**.

Blueprints: a machine readable model of the system itself



Model types

architecture · behavioral · conceptual · data flow · deployment · logical · network · operational · physical · process

Building blocks

assets, actors, zones, boundaries, flows, data stores, data sets, relationships, assumptions, use cases

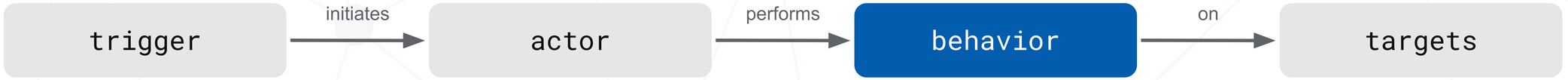
Visualizations

C4 · SysML · data flow · sequence · attack tree · and more

Threat models, risk analyses, and privacy analyses reference the blueprint rather than living inside it.

Behaviors: what things actually do

Behavior instance



An independent behavior taxonomy

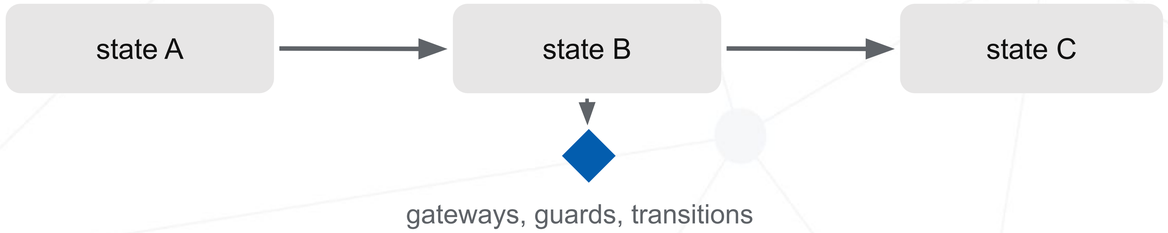
12 domains

89 categories

639 specific behaviors

AI · application · data · evasion · file · GUI · hardware · network · privacy · safety · security · system

Behavior graphs activity flows, state machines, or both



Every behavior is **declared**, **observed**, or both.

Act 2: Prove it

Substantiate every claim

From assertion to assurance

Assertion

“This is acme-lib 1.2.0.”

Trust the supplier

Evidence

Method, technique, confidence

How it was established

Assurance

“Here is how we know.”

Anyone can verify it

Assertion asks you to trust the list. Assurance shows its working.

In 2.0 the pattern applies to everything

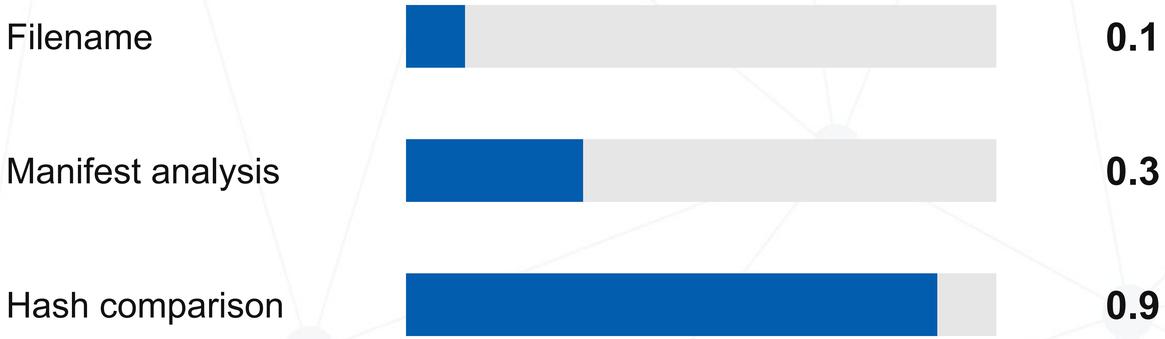
identity | vulnerabilities | threats | risk | controls | conformance | behavior

A claim says what you have. Evidence says how you know.

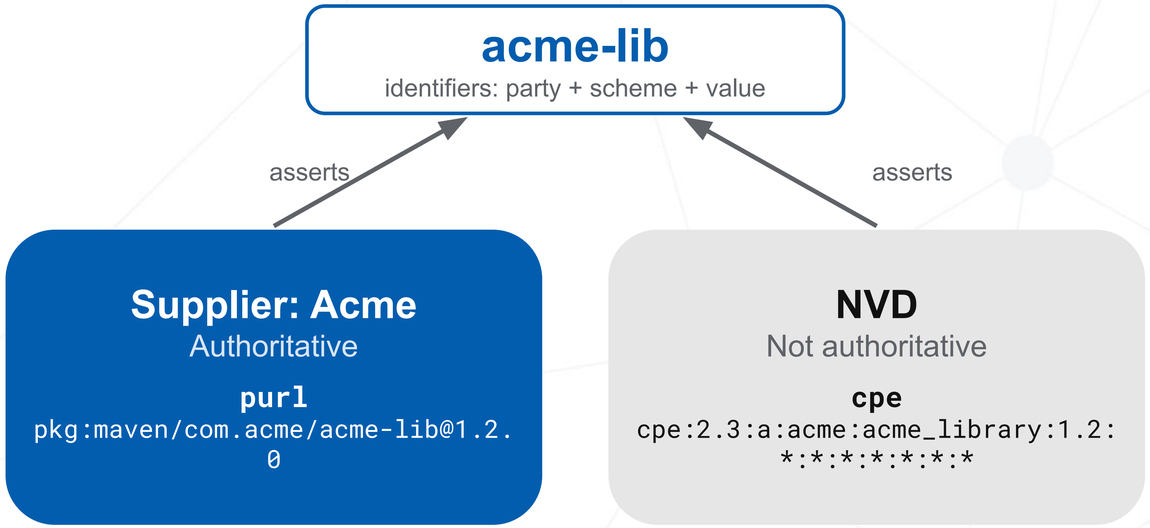
Concluded identity

`pkg:maven/com.acme/acme-lib@1.2.0`

Confidence by technique (0 to 1)



Identity is tied to who asserts it



Same component, two identities, each attributed to its source. Mismatches become visible.

Each technique carries its own confidence. A filename match is weak; a hash match is strong.

Evidence you can weigh

Every analysis method can record

`technique` and its confidence, from 0 to 1

`result` detected, not detected, or inconclusive

`rule` that produced it, and the tool that ran it

`supporting data` a photograph, a log, a screenshot, a request and response, a prompt and completion

`parties` involved, and their roles

Techniques now span

Software

source · binary · manifest · AST fingerprint · hash · content match · reachability · taint · instrumentation

Physical

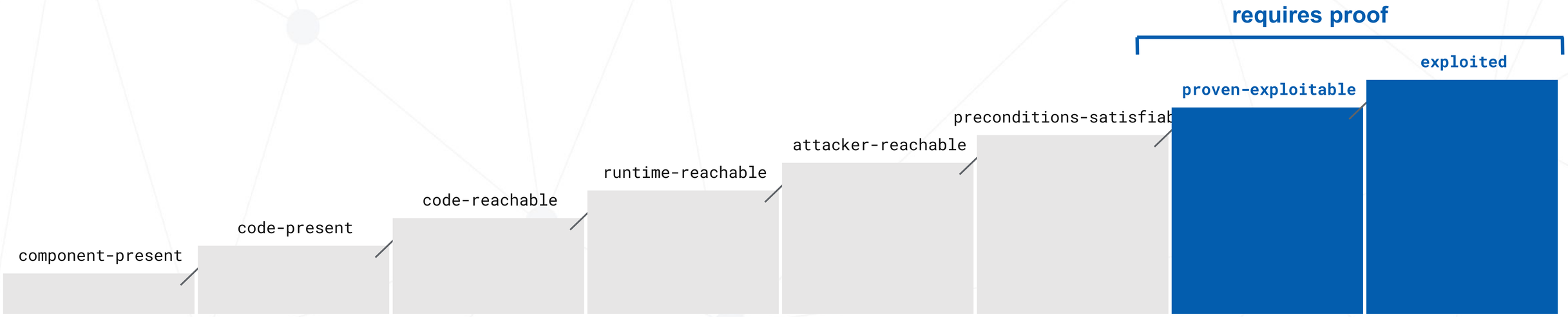
visual inspection · X-ray inspection · electrical testing · material analysis · decapsulation

Human

manual review · attestation

“Not detected” is meaningful negative evidence, distinct from a method that was never run.

Evidence with teeth



Exploitability is a ladder, and the top rungs require proof.

`proven-exploitable` and `exploited` fail validation unless backed by exploitation, penetration testing, dynamic analysis, instrumentation, or runtime protection.
EPSS is recorded as what it is: a probability, a percentile, and a rank, with the model version and timestamp.
VEX “protected by mitigating control” points at the actual control.

Threat modeling, natively

Methodologies

STRIDE | LINDDUN | PASTA | MAESTRO | OWASP | TRIKE | VAST | ATFAA | attack trees

`threat` **one model, many methodologies**

Categories from several taxonomies at once:

STRIDE LINDDUN MAESTRO layers MITRE ATT&CK tactics

If you say STRIDE, the category must be one of the six STRIDE values.

Building blocks

Scenarios	threats, threat scenarios, threat profiles, damage scenarios
Attack knowledge	attack patterns (CAPEC), techniques (ATT&CK)
Structure	attack trees, attack paths, abuse cases, trust boundaries, kill chain phases
Indicators	IoCs, IoAs, YARA, Snort, regex, hash, behavior signatures

A single threat can be categorized under STRIDE and MAESTRO at the same time, which matters for AI systems.

Cause, consequence, and the actor

cause: likelihood

Threat scenario

Who, why, and how feasible

actor, motivation, intent, access level, attack vector, likelihood

+

consequence: impact

Damage scenario

What harm results

affected subjects, impact per category

=

cause + consequence

Risk

A risk pairs a threat scenario with a damage scenario

likelihood paired with impact

Threat

What can go wrong

origin, categories, weaknesses, mitigations

Threat profile

What the actor can do, durably

sophistication, resources, skill set

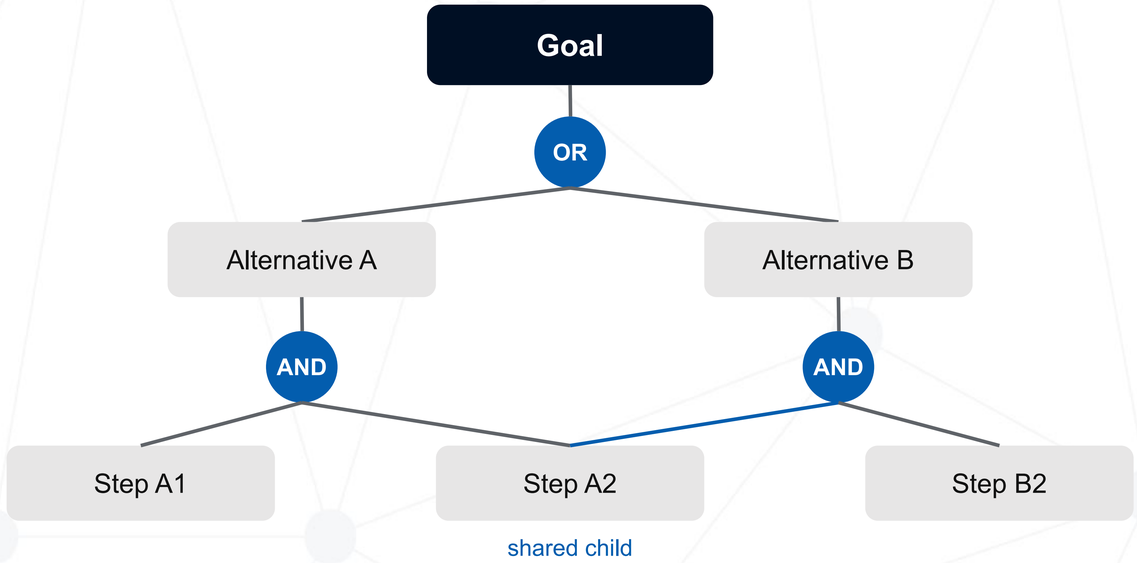
Actors are parties: named groups are organizations, attacker classes are personas.

Threat origins follow NIST SP 800-30: adversarial, accidental, structural, environmental.

Aligned with ISO 31000 and ISO/SAE 21434. One damage scenario can be paired with many causes.

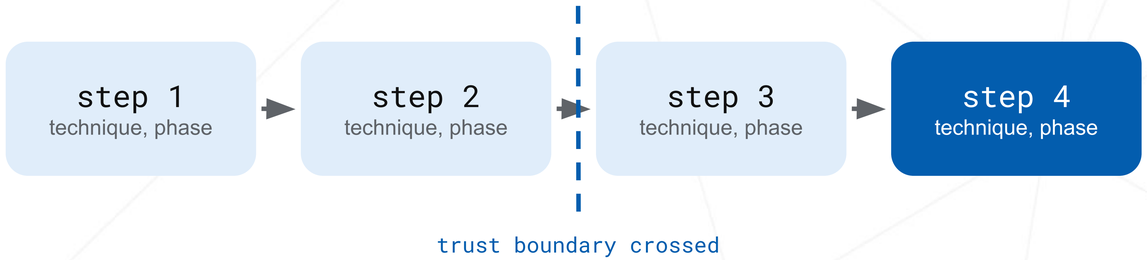
Attack trees and attack paths

Attack tree decompose a goal into alternatives



Flat list of nodes; each node lists children by reference with an and or or operator. A node can carry cost, skill, detectability, the attack pattern, the weakness exploited, and the mitigating controls.

Attack path the temporal progression of one attack



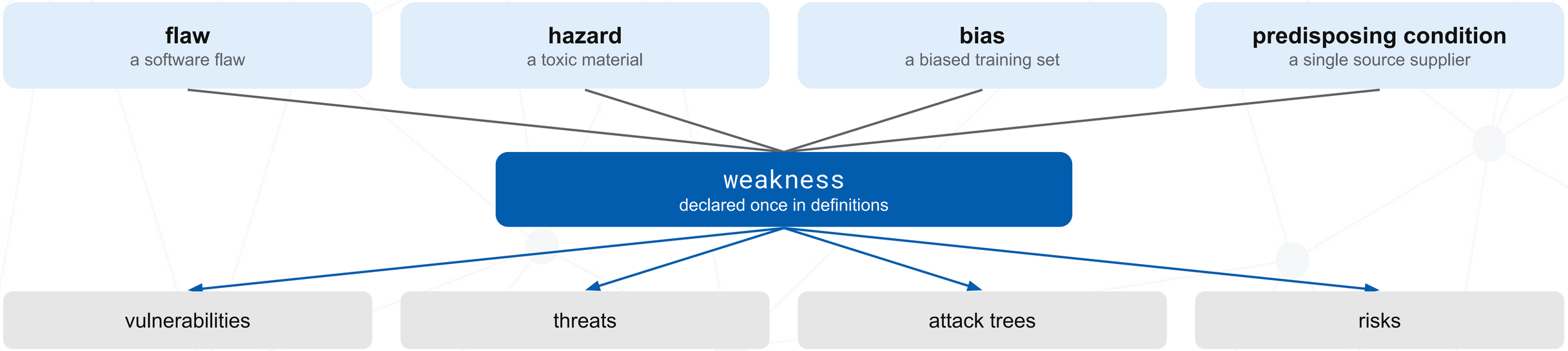
Ordered steps with technique, kill chain phase, source and destination, the trust boundary crossed, what was exploited, and which controls detect or prevent the step.

Abuse case the security counterpart of a use case

abuser, flow, targets, detection opportunities

Both reference the same blueprint boundaries and the same control objects, so mitigation coverage can be computed instead of argued.

Weaknesses: the source of risk



Schemes CWE, IMDRF adverse event terms, HFACS, MITRE System of Trust, GHS, ISO 6520, ISO/IEC 24772

Category taxonomies NIST SP 1270 and ISO/IEC TR 24027 for bias, NIST SP 800-30 for predisposing conditions

The consequence is not recorded on the weakness; it is recorded on the risk that references it.

Risk, in any framework

Bring your framework. CycloneDX can represent it.

Governance frameworks recorded as standards

ISO 31000 | NIST SP 800-30 | NIST AI RMF | COSO ERM | ISO/IEC 27005 | OCTAVE

`risk`

ratings in the same shape

qualitative

semi quantitative

quantitative

Scoring methodologies on each rating, with vector and scale

FAIR | OWASP Risk Rating | DREAD | FMEA | NIST SP 800-30 | OCTAVE Allegro | qualitative matrix | custom

Risk domains

security | privacy | safety | financial | regulatory | supply chain | ethical | fairness | human rights | environmental | and more

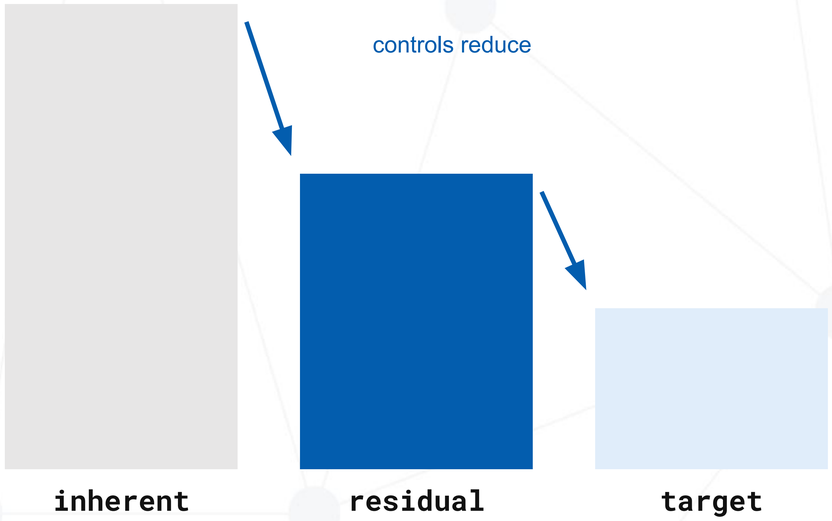
Anatomy of a rating

	Qualitative	Semi quantitative	Quantitative
Likelihood	very low to certain	score on method scale	probability, frequency, three point range
Impact	negligible to catastrophic	score	financial loss, currency, downtime, records, affected users
Score	negligible to critical	1 to 25 matrix, 0 to 9 OWASP, 1 to 1000 FMEA RPN	FAIR loss magnitude

Likelihood factors align with FAIR, OWASP, NIST SP 800-30, and ISO/IEC 18045 attack potential, each weighted, each with its reasoning.
A rating can carry several impacts, each a **harm or a benefit**. Confidentiality, integrity, and availability are recorded as the security attributes compromised.
Third axes when the method needs them: detectability for FMEA, controllability for ISO 26262.

Inherent, residual, target

Every risk separates three ratings, so reduction is explicit



Responses each names its controls: status, effectiveness, cost, owner, target date

avoid | reduce | share | accept | exploit | enhance

Status

`identified` ► `assessed` ► `treated` ► `monitored` ► `retired`

Risk appetite per party and per domain

averse | minimal | cautious | open | hungry

Assessments

DPIA, fundamental rights impact (EU AI Act), AI impact (NIST AI RMF), supply chain (NIST SP 800-161), model risk, safety, and more

Cadence includes **continuous** and **triggered**, with an explicit acceptable or unacceptable conclusion.

“Continuous, automated monitoring, as in a GRC engineering or continuous controls monitoring practice.”

Safety meets security

Safety integrity levels on assets

ISO 26262 ASIL | IEC 61508 SIL | DO-178C DAL | IEC 62304 class | EN 50128 SSIL

Example from the test suite: an infusion pump

TLS vulnerability
on an IEC 62304 class C asset

→ **safety impact assessment**
triggered

→ **VEX: not affected**
analysis

→ **conclusion: acceptable**
assessment

→ **attestation claim**
citing an FMEA excerpt

Feasibility

from ISO/IEC 18045 attack potential: elapsed time, equipment, knowledge

Controllability

as a rating axis for ISO 26262 hazard analysis

Behaviors

a risk can cite an observed behavior that defeats a declared safety behavior

References to hazard analyses, safety cases, test reports, and incident reports.

Security, safety, and assurance in one document.

Controls: inventory or prescription

`control` applies to
components, services, zones, boundaries

`control` implemented by
components, services, workflows, parties

`control` satisfies
requirements, including requirements inside standards, with OpenCRE mappings

Prescribed controls arrive as `recommended`

Category preventive | detective | corrective | compensating | deterrent | recovery

Effectiveness a percentage and a rating from ineffective to excellent

recommended asserts a control is needed; verified is backed by evidence. GRC as code.

Assert less. Substantiate more.

Everything can be substantiated through CycloneDX Attestations (CDXA)

Gaps point at controls through mitigation strategies.

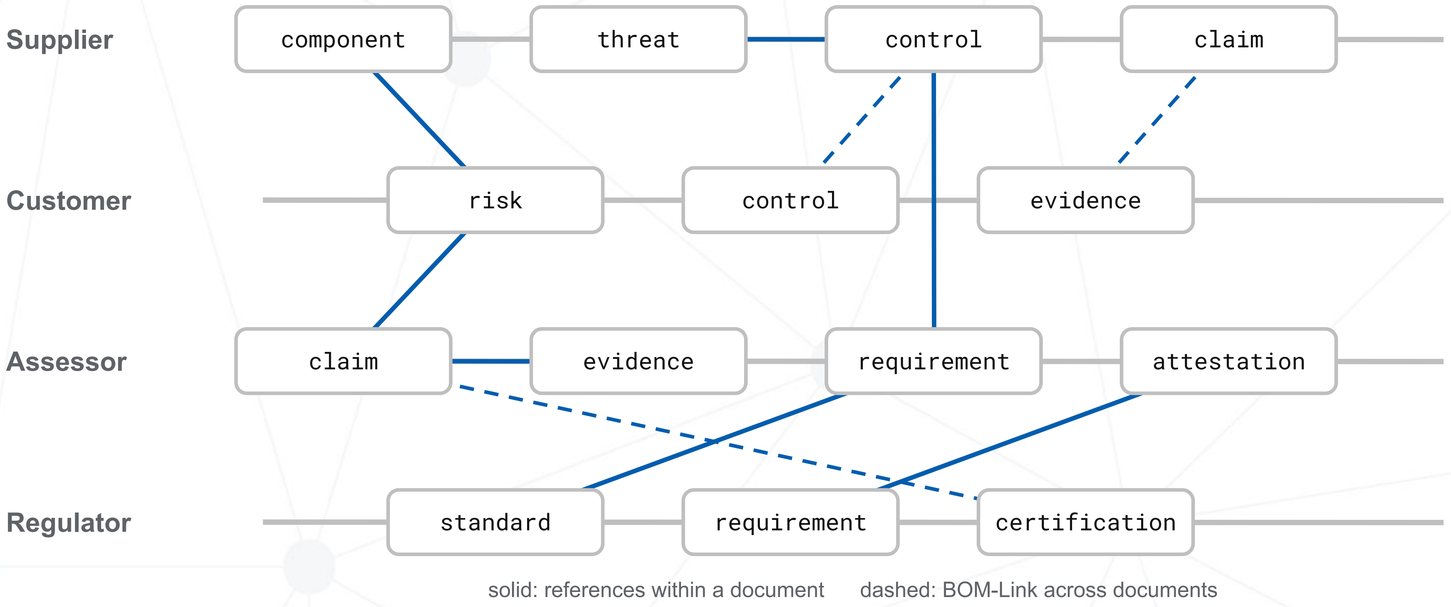
Risks and certifications point back at the claims behind them.

Signed by accountable parties (JSON Signature Scheme, ITU-T X.590). Evidence expires, which forces reassessment.

Act 3: The transparency fabric

One graph, every team, every organization

The transparency fabric



Primitives
every object has a refId

Links
references cross domains: threat to control to requirement; risk to claim to evidence

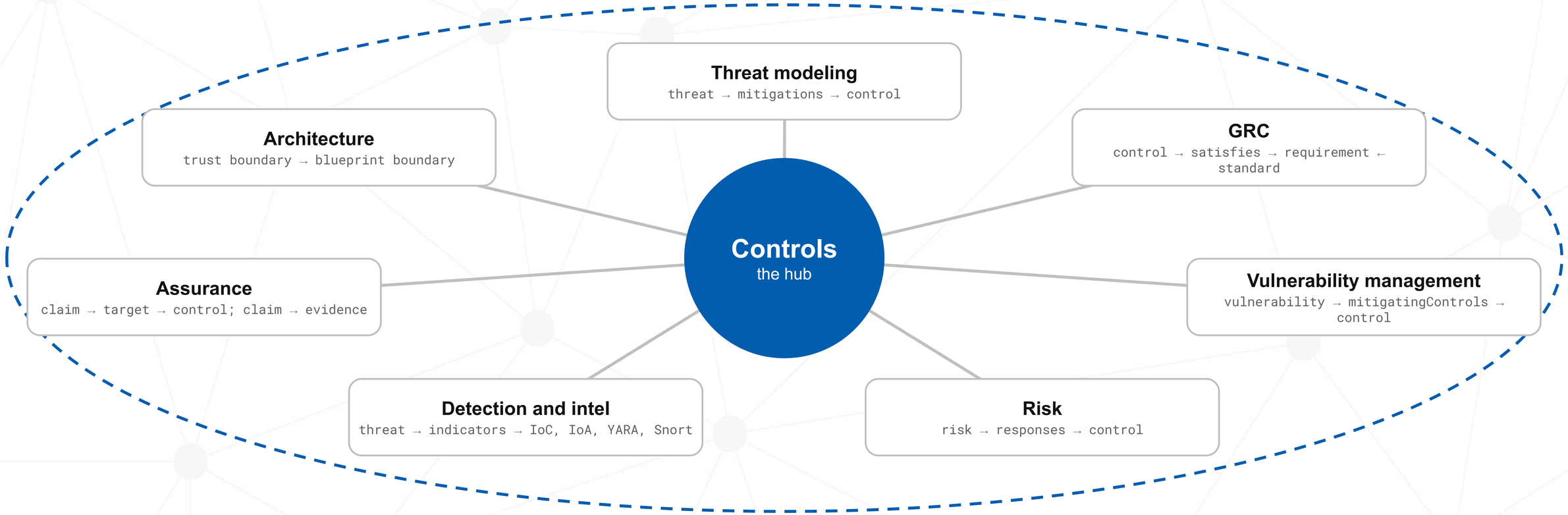
Across organizations
BOM-Link ties documents together: supplier, customer, assessor, regulator

Delivery
the same modules as a portable, signed file, or through an API for live exchange between tools

Views
perspectives let every team read the fabric in its own vocabulary

Reusable and linkable primitives

Breaking down silos



API the connective layer: tools exchange the same objects directly

Example: controls are the hub. Every team already touches them.

Perspectives: same data, native vocabulary

one signed document

`model properties`

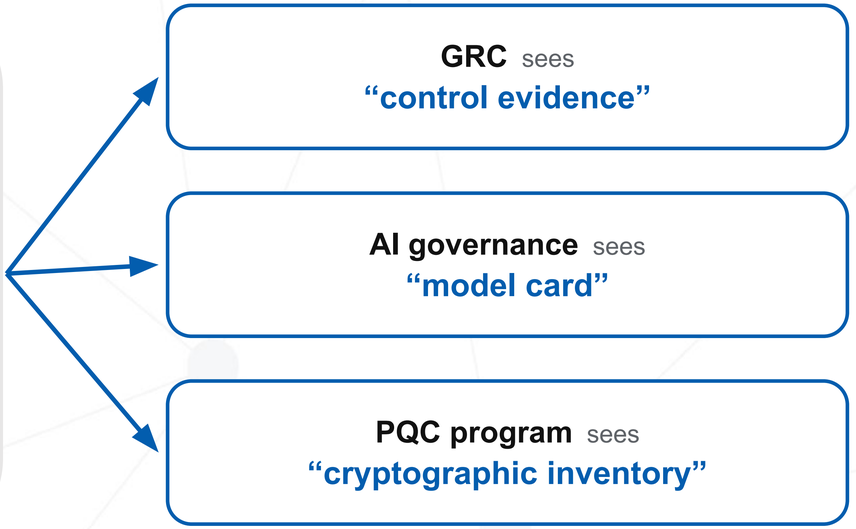
`measurements`

`risks`

`weaknesses`

`controls`

`cryptographic assets`



How a perspective works

JSONPath mappings select the data a team cares about

Native names translate CycloneDX terms into the team’s language

Relevance per mapping: required, recommended, optional, informative

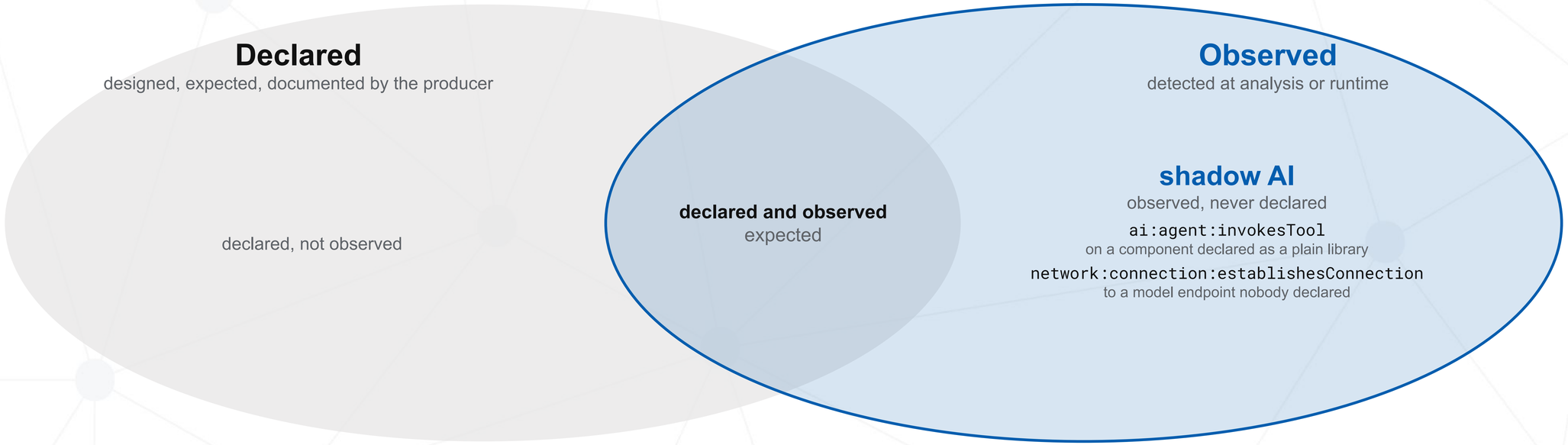
A perspective can also check completeness for a given audience.

80+ predefined domains, including

threat modeling | threat intelligence | risk management | regulatory compliance | automotive safety | medical devices | cryptographic security

Shadow AI: declared versus observed

Shadow AI is the gap between declared and observed behavior.



An observed behavior with no declared counterpart is a finding.

Threats reference the declared behaviors that constitute their intent. Risks reference behaviors directly.

Behaviors and cryptography

A CBOM tells you what algorithms exist. **Behaviors tell you what the system does with them.**

43

cryptographic behaviors

for example

Keys

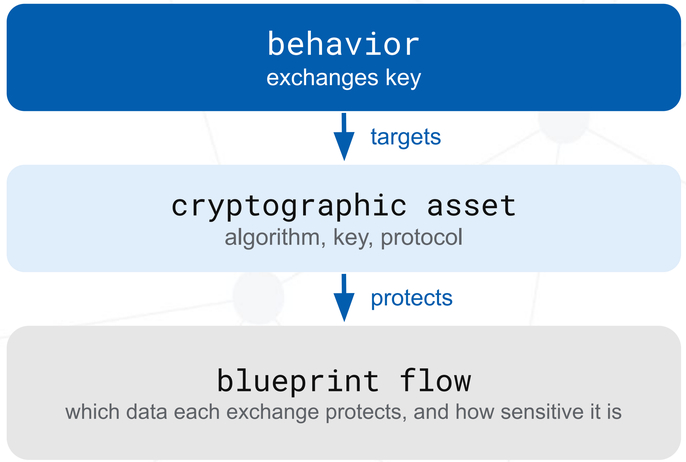
exchanges key | generates key | rotates key | wraps key | destroys key

Code and certificates

signs code | verifies code signature | validates certificate | checks revocation

Data

encrypts data in transit | encrypts data at rest



Harvest now, decrypt later is a behavioral risk: a key exchange using a classical algorithm on a flow that carries long lived confidential data.

Inventory finds RSA. Behavior plus blueprint tells you which uses of RSA protect data that must stay secret for a decade.

PQC readiness is agility

How was this configured?

hardcoded	product default	administrator configured	policy managed

hardcoded, not changeable → **negotiated, policy managed**

How can it change?

not possible	hardware replacement	firmware update	software update	restart required	runtime configuration

`nistQuantumSecurityLevel` from 0 to 6 per algorithm, plus parameter set identifiers (for example SLH-DSA under FIPS 205)

Is it negotiated at handshake?

Key rotation and certificate renewal: manual, on demand, automatic, via ACME, EST, CMP, KMIP, PKCS#11

Key states per NIST SP 800-57, including compromised

Readiness is not “do we use a quantum safe algorithm today.” It is “when we must change, can we, and how hard will it be.”

TC 54	Standardize core data formats, APIs, and algorithms that advance software and system transparency. TC54 standardizes OWASP CycloneDX along with proposals developed by the individual Technical Groups.	CycloneDX ECMA-424
TG1	Develop and maintain a standardized, format-agnostic API that enables the efficient discovery and exchange of Bills of Materials (BOMs) and other related artifacts and intelligence between systems.	Transparency Exchange API Project Koala
TG2	Develop and maintain the Package-URL specification. TG2 also standardizes the VERS specification for uniform version ranges and establishes an ongoing review process and governance for new PURL types.	Package-URL PURL and VERS ECMA-427
TG3	Develop the Common Lifecycle Enumeration (CLE) supporting component aliasing, component lifecycle events such as end-of-life (EOL) and end-of-support (EOS), and provenance chaining over time.	Common Lifecycle Enumeration ECMA-428
TG4	Develop a structured and machine-readable communication of an open source community's or project's state, needs, and support intentions. This includes codifying maintainers' support levels, signalling calls for assistance, identifying abandonment or handoff status, and improving visibility into whether a project is viable for commercial or long-term use.	Open Source Sustainability

Standardization Path

Get Involved

OWASP FOUNDATION

®

Participation

- OWASP initiatives are open to anyone
- Everyone has an equal seat at the table
- Output will be promoted to Ecma TC54

https://cyclonedx.org/participate/working-groups/

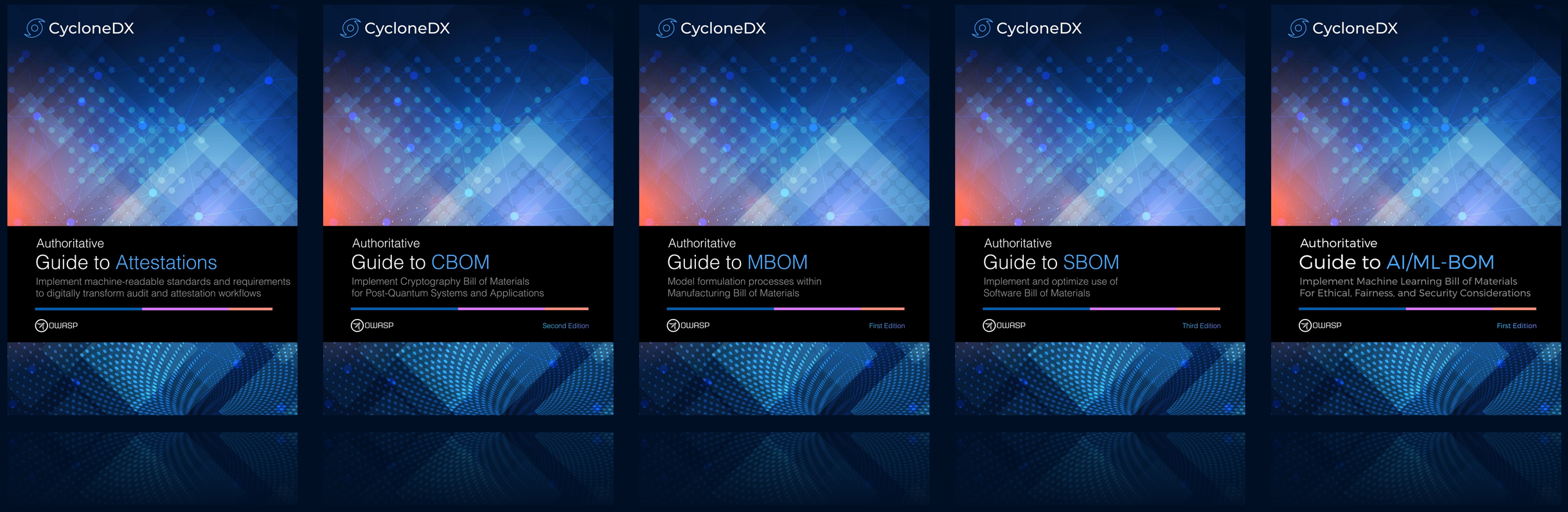
Participation

- Participation in the OWASP community open to everyone
- Participation in Ecma TC54 or any of the TGs requires:
  - Ecma membership
    - https://ecma-international.org/about-ecma/join-ecma/
  - Invited expert
  - Observer

Resources

- Ecma International TC54
  - https://ecma-international.org/technical-committees/tc54/
- TC54 Website
  - https://tc54.org/
- CycloneDX
  - https://cyclonedx.org
- Specification
  - https://github.com/CycloneDX/specification

Authoritative Guides



CycloneDX

Available now at: cyclonedx.org/guides

OWASP

Thank You

OWASP

TM