ARCO Project Defense: Official Script

Presenter: Alex Moskowitz **Role:** Chief Architect **Context:** Project 6 Defense (NCOR/Ontology)

SLIDE 1: Title & Scenario

*(**Action:** Stand tall. Do not look at the screen immediately. Look at the audience.)*

"Good afternoon. I am presenting **ARCO**, a compliance engine designed to solve a specific, high-stakes problem in the Defense sector.

Here is the operational reality: We deploy a fleet of perimeter security drones called **Sentinel-ID**. Under the new **EU AI Act**, specifically Article 6, any system capable of 'Remote Biometric Identification' is classified as **High Risk**.

The problem is **Latent Capability**.

Our drone hardware *has* a high-resolution camera. It *can* perform facial recognition. Even if we ship it with that feature turned off, the capability exists in the hardware. If a field operator flips a software switch, the legal classification of that drone changes instantly.

If we rely on static PDF manuals to track this, we are blind.

I built ARCO to solve this. It is a **Neuro-Symbolic Pipeline** that uses LLMs to read the messy documentation, but uses **Ontological Reasoning** to mathematically prove the compliance status."

SLIDE 2: Why We Created a Synthetic AI System

*(**Context:** Establish the "Safe Harbor" for testing.)*

"Before we dive into the logic, I need to clarify our testing methodology. We created a synthetic reference system called **Sentinel-ID**.

Why? Because real defense AI systems are black boxes. Their documentation is closed, and their hardware is proprietary. To prove that our compliance reasoning works, we needed a system where we could observe **everything**—every component, every capability, and every line of code.

Sentinel-ID serves as our 'Ground Truth.' It allows us to test the logic of the EU AI Act deterministically. We aren't guessing if a real vendor is non-compliant; we *know* Sentinel-ID is non-compliant, and we are using it to prove that our pipeline can detect that fact automatically."

SLIDE 3: System Overview (The Mermaid Graph)

*(**Action:** Walk them through the logic. Be precise about realization.)*

"This diagram represents the entire ARCO architecture in a single view. Let me teach you how to read it. There are three distinct layers working in unison:

1. **Reality (Bottom Left):** This is the Physical World (BFO). Here, we model the **Sentinel-ID System** and, crucially, its **Biometric Disposition**. This represents what the machine *is*.
2. **Regulation (Right Side):** This is the Legal World (IAO). We model the EU AI Act not as code, but as **Information Content** that is 'about' the system.
3. **Operations (Top Left):** This is the Process World. When the system runs a 'Surveillance Run,' that process *can* realize the biometric disposition, depending on configuration and context.

The **High-Risk Determination** (Center) is the bridge. It is an artifact that links the *System* to the *Regulation*. It isn't a label; it's a relationship."

SLIDE 4: Why the EU AI Act Must Be Modeled, Not Coded

*(**Context:** Preempt the "Process vs. Plan" critique.)*

"A key engineering decision was how to handle the Law. The EU AI Act is written for humans. It describes categories and conditions, not executable procedures.

If you try to write Python `if/else` statements for Article 6, you will fail. The law is ambiguous.

Instead, we treated the Law as a **Reference Model**. We represented the Act as **Information Content Entities** in the graph. The evaluation itself is a separate **Process** that takes the law as a specified input and the system as a participant.

Our system doesn't 'execute' the law; it refers to the law as a constraint."

SLIDE 5: Why There Are Two EU AI Act Models

*(**Action:** Teach the graph reading here.)*

"We actually built **two** separate models for the Act, because the Act does two different things.

- **Model 1 is Structural:** It answers, 'What makes a system High Risk?' (Article 6).
- **Model 2 is Governance:** It answers, 'Who has the authority to change the rules?' (The EU Commission).

Before I explain why we separated them, a quick note on reading these graphs: **Whenever you see `rdf:type`, that is the point where we assert that this specific instance—this drone—is an instance of a universal class defined in the ontology.**"

SLIDE 6: The Governance Graph (Deep Dive)

*(**Action:** Gesture to the top nodes.)*

"This is the Governance Model. Here we see the **EU AI Board** and **European Commission** modeled as Organizations bearing Authority Roles. These roles participate in processes that output **Delegated Acts**.

We separated this from the classification logic so that the *Governance* can evolve—new boards, new processes—without breaking the core *Classification* logic. This ensures the system is maintainable over decades, not just weeks."

SLIDE 7: The Classification Graph (Deep Dive)

"And this is the Classification Model. This graph captures the logic of **Article 6**. You can see 'Annex III List' and 'Classification Criteria' modeled as Information Content.

The arrows here are critical. They are `is_about` relations. The Classification Criteria are *about* the System. This allows us to query the graph and ask: *'Which specific clause of the law is this drone violating?'*"

SLIDE 8: Formal Regulatory Source Model

*(**Context:** Summarize the regulatory stance.)*

"To summarize the Regulatory Layer: We treat the EU AI Act as a **Constraint**.

The Regulation exists in the graph as a standard against which the System is measured. A 'High-Risk Determination' is only generated if the System's facts align with the Annex III conditions modeled here."

SLIDE 9: ARCO System Overview (Text)

"Bringing it back to the System View: We have the **Realist Core** (The Drone). We have the **Regulatory Layer** (The Law). And we have the **Governance Extension** (The Provider).

This V3 Governance extension is critical. It links the **System** to the **Provider Organization**. This means if a drone is found non-compliant, we don't just flag the machine; we identify the legal entity responsible for it."

SLIDE 10: The Interpretability Gap

*(**Action:** Narrative Anchor. Hammer the problem home.)*

"Let's step back. Why did we need all this ontology? Why not just use Python? Because of the **Interpretability Gap**.

On the left, you have the Law: 'Biometric,' 'Remote,' 'Public.' These are vague universals. On the right, you have the Tech: Firmware versions, sensor logs.

Traditional ETL pipelines break here. You cannot regex 'Regulatory Intent.' And purely Generative AI is dangerous. An LLM might hallucinate compliance because the marketing brochure looks nice.

Latent capability creates latent liability. We needed a bridge."

SLIDE 11: Architecture & Project Alignment

*(**Context:** Address the OWL/SHACL logic.)*

"This architecture satisfies the **Project 6 Rubric** for Design Pattern Expansion. It is a 3-Stage Pipeline:

1. **Interpretation (Neuro):** The LLM acts as the eyes. It reads the text.
2. **Representation (Symbolic):** The Ontology acts as the brain. It structures the data.
3. **Reasoning (Logic):** The Reasoner acts as the judge. It proves the classification.

Because this is OWL, we operate under the **Open World Assumption**. We use SHACL earlier in the pipeline to ensure the required facts are present before any reasoning occurs."

SLIDE 12: LLMs as Candidate Generators

*(**Action:** Point to the "No Direct Touch" box.)*

"This is the most important architectural rule: **LLMs are Candidate Generators, not Judges.**

The LLM scans the documentation and proposes: *'This system might be biometric.'* We treat this as a **Noisy, Probabilistic Hypothesis**.

We filter it through the Ontology. If the LLM proposes a relationship that violates BFO constraints, we reject it. The LLM never touches the final compliance conclusion. It only offers candidates."

SLIDE 13: Grounding Risk in Reality

*(**Action:** Slow down. This is the "Aha!" moment.)*

"Here is how we grounded the risk. We modeled Risk as a **Disposition**.

In BFO, a Disposition is a capability that exists in the hardware even when it is not active. This solves the 'Software Switch' problem. Because the drone *bears* the disposition for facial recognition, it is High Risk *by nature*, not just by *deployment*.

We are judging the machine's physics, not just its settings."

SLIDE 14: Deterministic Inference

"Finally, the Logic Layer. We encode the **regulatory constraint** as an OWL Restriction.

IF System bears_disposition Biometric, THEN HighRisk.

When we run the HermiT reasoner, it provides a **Logical Necessity**. If the premises are true, the conclusion *must* be true. This is not a probability score. It is a mathematical proof."

SLIDE 15: Operational Validation

"To make this production-ready, we added two safety nets:

1. **SHACL:** Validates the structure. It stops bad data before reasoning begins.
2. **SPARQL:** Provides traceability. It allows us to query the graph and extract the exact audit trail."

SLIDE 16: Why Glass-Box Compliance Scales

"This isn't just about drones. This pattern scales.

- **Healthcare:** Linking diagnostic capabilities to clinical approval.
- **Finance:** Linking trading algorithms to regulatory bounds.
- **Defense:** Linking classified capabilities to authorized use.

Anywhere you have **Latent Capability** colliding with **Regulation**, this Neuro-Symbolic pattern applies."

SLIDE 17: Conclusion

"In conclusion, ARCO demonstrates that we don't have to choose between the flexibility of LLMs and the rigor of Ontology.

By architecting them together, we move from 'Black-Box' prediction to 'Glass-Box' compliance. We are no longer asking *if* an AI is compliant. We are **proving** it.

Thank you."