Project Abstract

Modern cybersecurity architectures are fundamentally reactive, suffering from a critical detection gap: the *inability to observe and analyze post-compromise attacker behavior*. While perimeter defenses have matured, sophisticated adversaries routinely breach organizations and operate *undetected for months*, executing lateral movement, privilege escalation, and data exfiltration in what amounts to a security blind spot. This detection deficiency stems from traditional security tools' focus on preventing entry rather than managing attackers who have already penetrated defenses.

We present **Maya**, an **autonomous deception framework** that fundamentally reimagines enterprise defense by shifting from passive detection to active adversarial engagement. Unlike conventional security measures that attempt to block attackers at the perimeter, Maya creates a **state-synchronized parallel reality**, a complete, believable alternative infrastructure that attackers are systematically lured into and contained within. This framework transforms compromised footholds from security liabilities into intelligence assets, providing defenders with unprecedented visibility into adversary behavior.

The Core Innovation:

Maya introduces a **CRDT-based state synchronization engin**e that maintains *persistent, consistent attacker identity and privilege context* across many interconnected deception nodes. This creates what we term "**adversarial continuity**", the illusion of moving through real infrastructure while every action is instrumented, analyzed, and contained. The system **dynamically adapts deception fidelity** through machine learning algorithms that adjust responses based on observed attacker sophistication, creating a personalized engagement that maximizes intelligence capture while preventing detection of the deception layer.

What Makes Maya Unique:

- **Cohesive Deception Ecosystem:** While existing solutions deploy isolated honeypots, Maya creates an entire parallel infrastructure with interconnected services, shared credentials, and logical data flows that mirror actual organizational environments.
- **Behavioral Adaptation Engine:** The system employs reinforcement learning to adjust deception responses in real-time, creating progressively engaging environments that adapt to attacker skill levels: novice attackers receive simpler decoys while APT-level adversaries encounter sophisticated, multi-layered deception.
- **Proactive Intelligence Generation:** Unlike traditional security tools that generate alerts, Maya produces actionable threat intelligence with MITRE ATT&CK mapping, complete attack narratives, and attacker behavioral profiles that enable proactive defense.
- **Infrastructure-Agnostic Deployment:** The framework operates transparently alongside production systems, requiring no architectural changes while providing comprehensive

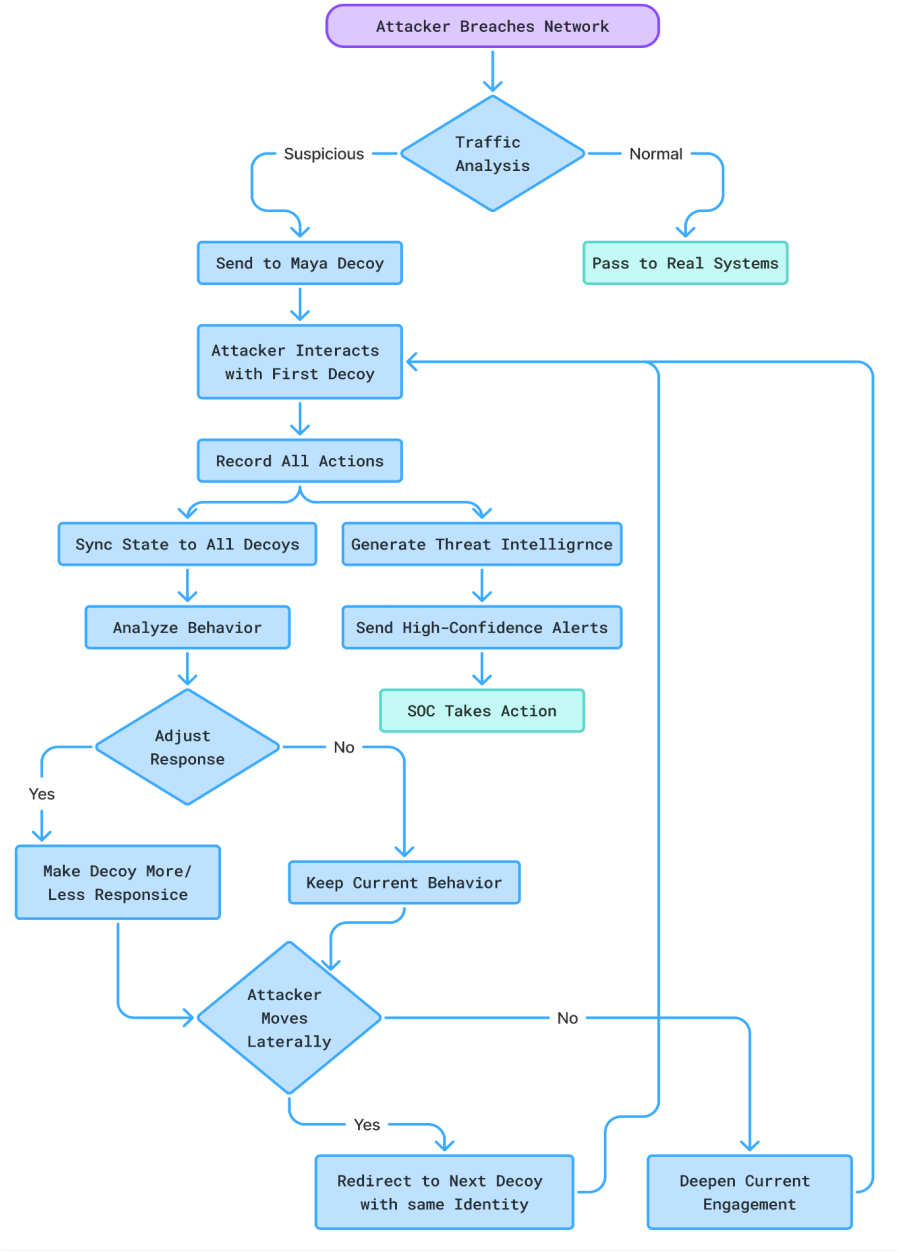
coverage through multiple integration methods (DNS, credential, network, and service discovery).

Technical Differentiators:

- **State Synchronization:** Conflict-free replicated data types ensure attacker context persists across all deception nodes, creating a seamless experience that resists fingerprinting.
- **Dynamic Content Generation:** AI-powered systems automatically create decoys that precisely match organizational environments, including custom applications, data structures, and user behaviors.
- **Multi-Vector Luring:** Strategic placement of breadcrumbs: fake credentials, decoy DNS records, network artifacts, and file traces, ensures attackers encounter deception regardless of entry point.
- **Progressive Engagement Tiers**: Three levels of decoy interaction (low, medium, high) provide appropriate engagement based on attacker behavior while minimizing resource consumption.

Impact and Applications: In controlled deployments, *Maya extends attacker dwell time by 300-500% compared to traditional detection methods*, providing security teams with **early warning** and **comprehensive behavioral intelligence.** The system generates zero false positives: every interaction represents genuine malicious activity, dramatically reducing SOC alert fatigue. By transforming initial compromises into intelligence collection opportunities, Maya enables organizations to shift from reactive defense to proactive adversary understanding, ultimately strengthening security postures through empirical attack data.

Attack Flow with Maya Deception Layer



Step-by-Step Flow:

1. Initial Breach & Traffic Analysis
    - Attacker gains initial access (phishing, leaked creds, exploit)
    - Incoming traffic is analyzed in real-time
    - Decision point:
        - **Normal behavior : Redirect to Real systems**
        - **Suspicious behavior : Redirect to Maya decoy fabric**

2. First Decoy Interaction
    - Attacker lands on an initial decoy (SSH, web app, DB, AD)
    - Decoy behaves realistically
    - No obvious red flags or delays

3. Full Instrumentation

    Once inside:
    - Every command
    - Every credential use
    - Every file touch
    - Every lateral movement attempt

    is:
    - Recorded
    - Timestamped
    - Identity-linked
    - Shared across the deception network

4. Lateral Movement Handling

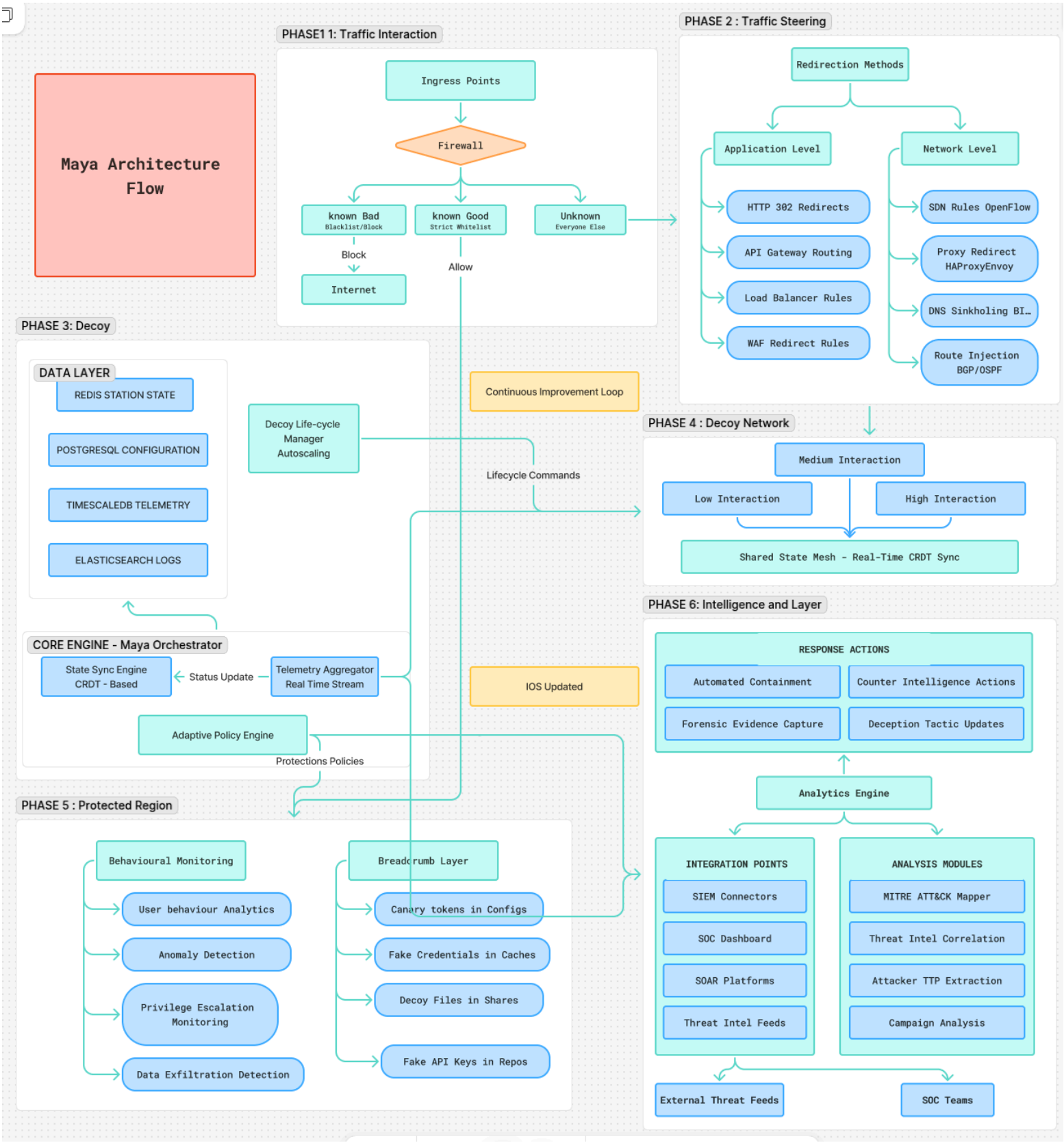
    If attacker tries to pivot:
    - Maya **redirects them to another decoy**
    - Same credentials still work
    - Same permissions persist

5. Adaptive Engagement Loop

    Maya continuously:
    - Analyzes attacker behavior
    - Adjusts decoy responsiveness
    - Decides whether to:
        - Deepen engagement
        - Keep behavior stable
        - Escalate alerts to SOC

SOC only receives **high-confidence intelligence**, not noise.

Project Architecture:



Phase 1: Traffic Comes In

When someone tries to connect to your company's network, Maya first figures out who they are:

- Known Good People (Employees): Go straight to real systems
- Known Bad People (Blocked hackers): Get stopped immediately
- Unknown People (Everyone else): Get sent to the **Maya Honeynet**

Phase 2: Redirecting Traffic

Once we decide someone goes to the fake world, Maya honeynet uses two ways to get them there:

Network Level:

- Changes network rules behind the scenes
- Redirects internet traffic secretly
- Changes DNS (website addresses) to point to fake servers

Application Level:

- Changes web page redirects
- Routes API calls to fake services
- Makes load balancers send traffic to decoys

Phase 3: The Control Center (Maya's Brain)

This is where everything gets coordinated:

Data Storage:

- *Redis*: Remembers what hackers are doing right now
- *PostgreSQL*: Stores all the system settings
- *TimescaleDB*: Records every single action taken
- *Elasticsearch*: Keeps detailed logs of everything

Core Engines:

- *State Sync Engine*: Makes sure if a hacker steals a password in one fake system, it works in all the others too
- *Telemetry Aggregator*: Collects all the information about what hackers are doing
- *Decoy Lifecycle Manager*: Creates and manages all the fake systems automatically

Phase 4: The Fake World (Decoy Network)

This is the actual fake environment with three levels:

- *Low Interaction*: Simple traps that just detect someone's there
- *Medium Interaction*: Interactive systems that let hackers explore a bit

- *High Interaction*: Complete fake computers that look and feel 100% real

The smarter the hacker acts, the more real the fake world becomes.

Phase 5: Protecting the Real Stuff

While hackers play in the fake world, Maya honeynet also watches your real systems:

- Tracks what employees normally do
- Spots unusual behavior
- Watches for privilege escalation attempts
- Detects data theft patterns

Phase 6: The Intelligence System:

Analysis Engine :

- Maps hacker actions to known attack patterns (MITRE ATT&CK)
- Correlates information with threat intelligence
- Extracts hacker techniques and procedures
- Analyzes entire attack campaigns

Response Actions:

- Automated Containment: Automatically isolates threats
- Counter Intelligence: Fights back against hackers
- Forensic Evidence: Collects proof of everything
- Deception Updates: Makes the fake world even better based on what we learn

Connections to Your Team:

- SIEM Connectors: Sends alerts to your security tools
- SOC Dashboard: Shows everything happening in real-time
- SOAR Platforms: Automates responses
- Threat Intel Feeds: Gets updates about new hacker methods

The Continuous Improvement Loop

The most important part: Maya honeynet gets smarter every day. Every time a hacker gets caught, the system learns:

- Updates the bad actors

- Improves the fake world to be more convincing
- Shares intelligence with your security team
- Adapts to new hacker techniques

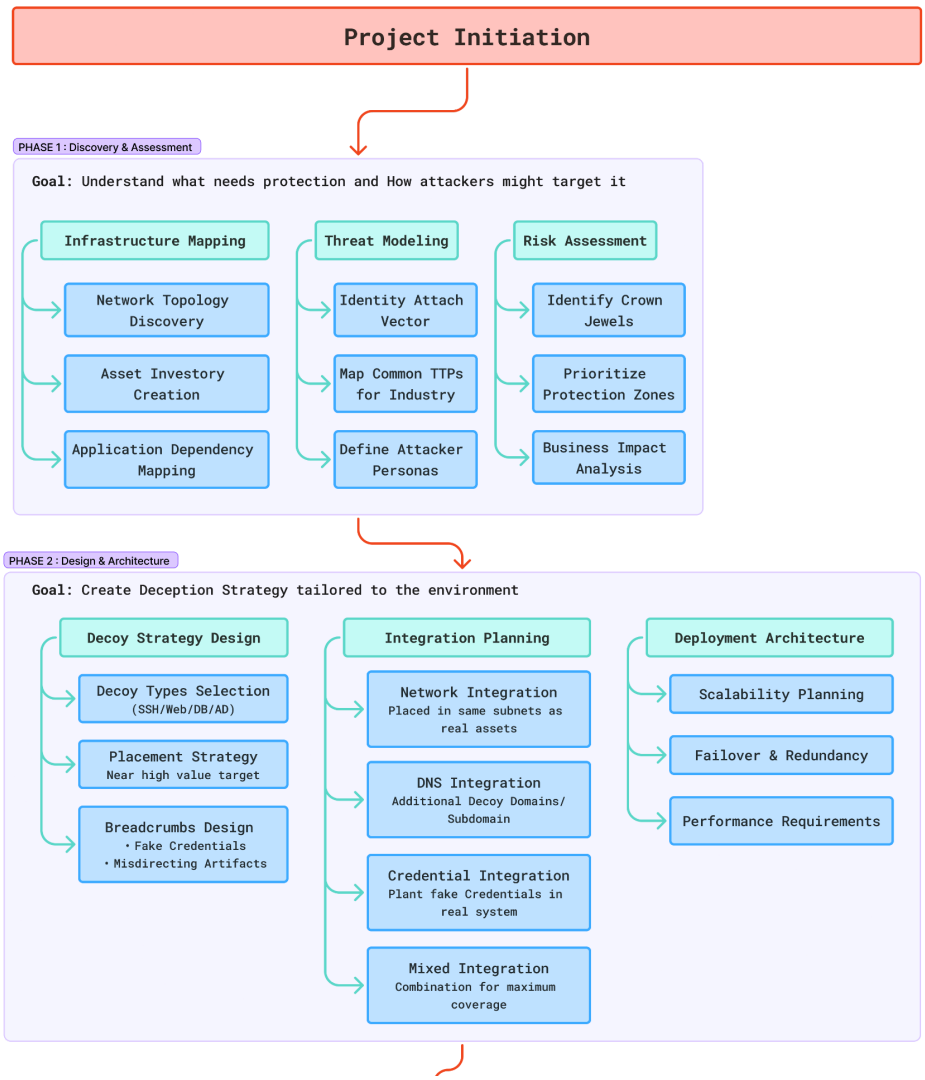
How Maya can be Integrated with Organization's Infrastructure:

Phase 1: First, We Learn Your Company

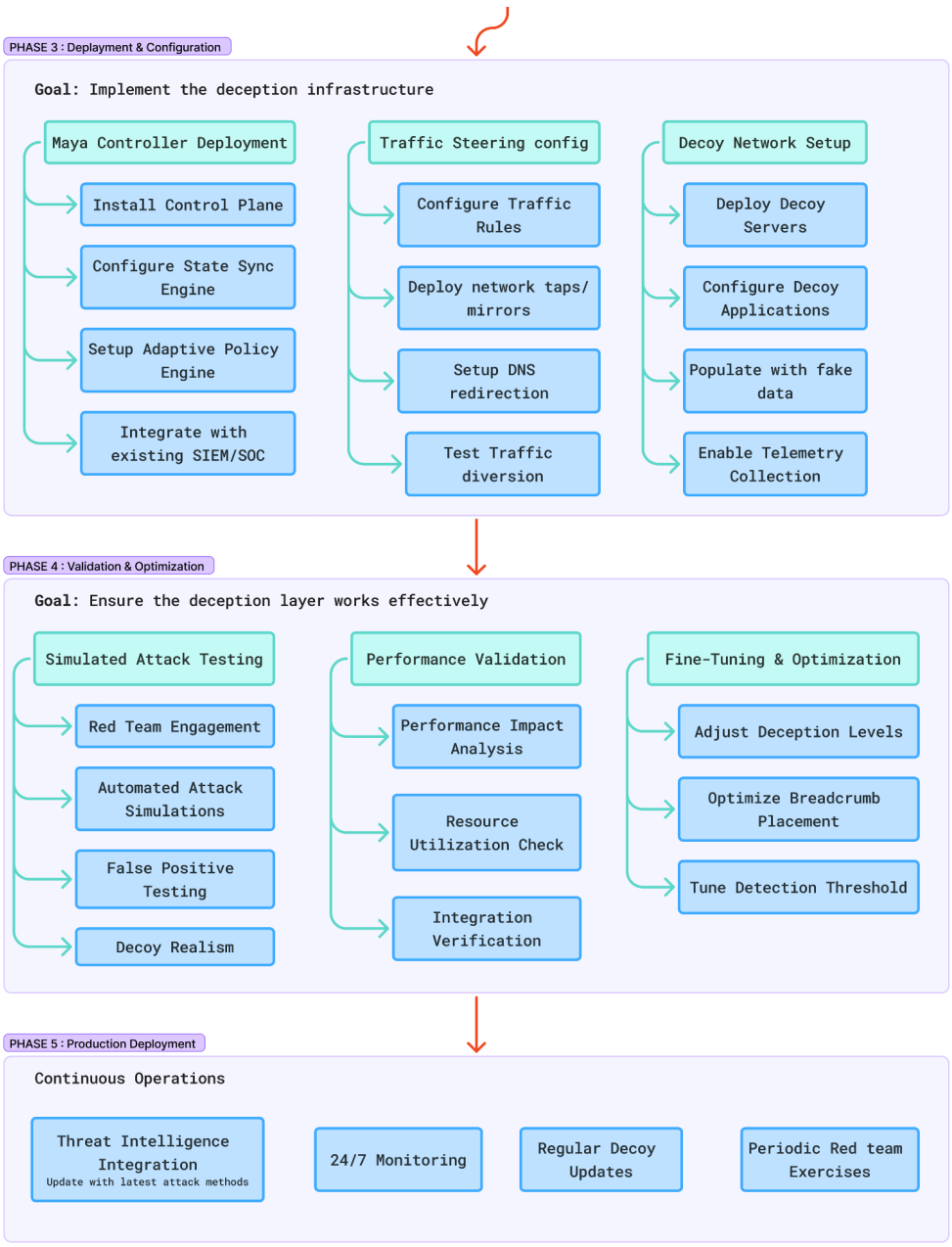
- We start by understanding exactly what your company looks like digitally:
- We map out all your computers, servers, and networks
- We identify what's most important to protect (your "crown jewels")
- We figure out how hackers might try to attack you
- We understand what normal looks like for your company

Phase 2: Then We Design the Fake World

- Based on what we learned, we create your fake digital twin:
- We build fake versions of your important systems



- We create fake passwords and documents (we call these "breadcrumbs")
- We decide where to place everything so hackers will find it
- We make sure the fake world can grow with your company



Phase 3: Now We Build Everything

- This is where we install all the parts:
- The Control Room: Where your security team can watch everything
- The Redirect System: Automatically sends suspicious people to the fake world
- The Fake Systems: Servers, websites, and databases that look real

- The Brain: Makes the fake world adapt based on what hackers do

Phase 4: We Test Everything Thoroughly

- Before going live, we make sure it all works perfectly:
- We hire friendly hackers to try to break in (they know it's a test)
- We check that real employees aren't affected
- We make sure the fake systems look and feel completely real
- We fix anything that doesn't work quite right.

Phase 5: We Go Live and Keep Improving

- Once everything works, we turn it on and keep it running:
- Real hackers get trapped in the fake world
- Your security team gets alerts with complete evidence
- We keep the fake world updated with new hacker tricks
- We regularly test it to make sure it stays convincing