NetTower – Sprint 1 Review

Benjamin Molloy
ASE Capstone
March 2, 2026

Sprint 1: Backend Discovery & Modeling Complete

Problem Reminder

Small or disrupted networks lack simple situational awareness tools.

- Enterprise tools:
  - Too heavy
  - Too complex
  - Infrastructure dependent

NetTower Goal: Provide lightweight, deployable network topology awareness.

Sprint 1: Scope & Completion

Sprint 1 Focus: Backend Discovery & Relationship Modeling

Planned Scope

- Establish project architecture
- Implement agentless discovery
- Model hosts and inferred relationships
- Stabilize backend pipeline
- Complete working prototype of back end

Completed Work

- Core backend architecture established
- Event models & EventBus system implemented
- MongoDB integration complete
- Passive + active discovery integrated
- Host correlation logic implemented
- Connectivity (edge) modeling functional
- Enrichment (OUI + device hints) added
- Orchestrator loop stabilized

Milestones & Deadlines Met

- Milestone 1 – February 16, 2026
  - Planning & Backend Architecture Finalized
- Hard Deadline 1 – March 9, 2026
  - Backend Core Complete (Sprint 1 Presentation)

Sprint 1: Burn Down

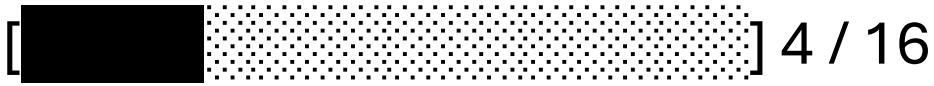
Burn Down

- Total Defined Requirements: **16**

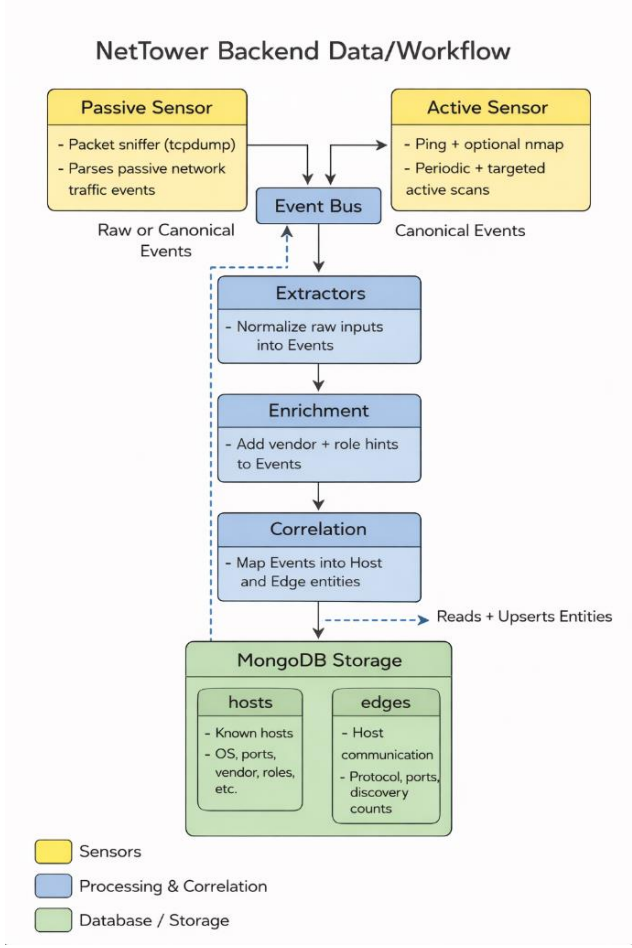
- **Backend-Satisfied Requirements: 4 / 16**
  - ~25% Complete

- Remaining Requirements: **12**
  - Remaining Focus: Visualization, interaction, UI integration (Sprint 2 focus)

Total Lines of Code: 2,986



Sprint 1: Demo Part 1




```
NetTower/
└─ code/
   ├─ frontEnd/
   │  ├─ app.py
   │  ├─ templates/
   │  │  └─ index.html
   │  └─ static/
   │     ├─ css/
   │     └─ js/
   │
   └─ backEnd/
      ├─ __init__.py
      ├─ main.py
      ├─ config.py
      ├─ settings.py
      │
      ├─ models/
      │  ├─ __init__.py
      │  ├─ events.py
      │  ├─ entities.py
      │  └─ types.py
      │
      ├─ pipeline/
      │  └─ event_bus.py
      │
      ├─ sensors/
      │  ├─ passive_listener.py
      │  └─ active_discovery.py
      │
      ├─ processors/
      │  ├─ extractors.py
      │  ├─ enrichment.py
      │  └─ correlation.py
      │
      ├─ storage/
      │  ├─ mongo_client.py
      │  └─ librarian.py
      │
      ├─ resources/
      │  ├─ oui_vendors.json
      │  └─ device_hints.json
      │
      └─ utils/
         ├─ logging.py
         └─ net.py
```

Sprint 1: Demo Part 2

WILL PUT DIAGRAM HERE

Learning with AI - Packet Structure & Protocol Breakdown

Objective: Understand packet behavior from frame header to payload.

Studied:

- Ethernet, IP, TCP, UDP, ICMP headers
- Flags and response codes
- What indicates reachability vs filtering

AI Used For:

- Breaking down protocol mechanics step-by-step
- Explaining open vs closed vs filtered behavior
- Clarifying what packet signals truly indicate

Impact on NetTower:

- Improved interpretation of probe results
- Reduced incorrect assumptions about reachability

Learning with AI - L2, L3, L4 Behavior, Constraints & Design Integration

Objective: Understand what Layers 2–4 expose — and what they do not.

Studied:

- Effects of NAT, firewalls, segmentation
- Timeouts vs rejects
- Visibility limits in constrained networks

AI Used For:

- Exploring misleading signals
- Stress-testing inference assumptions
- Validating cautious reasoning

Impact on NetTower:

- Clearer distinction between confirmed vs inferred conclusions
- More accurate uncertainty representation in visualization design
- Stronger architectural decisions grounded in protocol behavior
- AI supports technical learning — not product features.

Risks Identified

- Incomplete or misleading network data
- Overconfidence in inferred relationships
- Segmented or filtered environments
- Time constraints

Mitigation:

- Explicit uncertainty modeling
- Incremental architectural validation

Sprint 2 Plan

- Sprint 2 Focus: Frontend, Visualization Documentation and Testing
  - Build topology interface
  - Integrate backend API
  - Implement interactive navigation
  - Add heat mapping & visual differentiation

- Goal: Functional prototype (Backend + Visualization)

Sprint 2 Schedule:

- S2W1 (Week 10 – 3/16) Building Front End
- S2W2 (Week 11 – 3/23) Refinement of front end and Visual enhancement
- S2W3 (Week 12 – 3/30) Frontend–Backend Integration & Topology Rendering
- S2W4 (Week 13 – 4/6) Performance Optimization & System Refinement + User Testing
- S2W5 (Week 14 – 4/13) Documentation and publication prep
- S2W6 (Week 15 – 4/20) Final testing, demo, and submission

Closing Summary

Sprint 1 Outcome:

- Backend discovery pipeline operational
- Host and relationship modeling functional
- Architecture stabilized
- Technical assumptions validated through AI-supported learning

Next: NetTower is ready for visualization development in Sprint 2.

Questions?