SpeakUp Project

Customer Briefing Meeting Agenda

Bruce Dombrowski

January 6, 2026

Repository: `https://github.com/brucedombrowski/SpeakUp`

All artifacts, source code, and verification evidence available for review

Generated: January 6, 2026 09:18 UTC

1. Meeting Overview

Duration: 45–60 minutes
Format: Presentation + Live Demo
Audience: Technical leadership, program managers, customers

Meeting Objectives:

- Demonstrate a systems-engineering approach to knowledge work
- Show working Bundle Protocol (BPv7) implementation with live network capture
- Present workflow model for improved productivity and traceability
- Provide verification evidence and compliance documentation

2. Agenda

Time	Topic	Description
5 min	Introduction	Project genesis and governing principles
10 min	Problem Statement	Eight systemic constraints limiting effectiveness
10 min	Proposed Solution	Workflow model: Ideation → Execution → System of Record
15 min	**Live Demo**	Wireshark capture of BPv7 bundle transmission
5 min	Verification	Security scans, compliance evidence, traceability
5 min	Value Proposition	One person, entire team's output
5 min	Next Steps & Q&A	Discussion and path forward

3. Key Files to Share

Before the meeting, have these ready:

File	Purpose
`README.md`	Authoritative project specification and requirements
`briefing/SpeakUp-Briefing.pdf`	Executive briefing deck (main presentation)
`verification/Compliance-Statement.md`	Information handling verification
`verification/Requirements-Traceability.md`	Requirements-to-evidence matrix

For the live demo, have these ready:

File	Purpose
`src/bpv7/capture_demo.sh`	Wireshark capture demo script
`src/bpv7/test_local.py`	Console-based bundle test with hex dump
`src/bpv7/test_pdf_transfer_tshark.py`	PDF transfer with tshark verification

4. Presentation Script

4.1 Introduction (5 minutes)

Key talking points:

> "SpeakUp is a response to two calls—our organization's call for employees to speak up constructively, and our customer's request for process improvement recommendations.
>
> This project demonstrates that **with the right environment, one person can do the work of an entire team**. Everything you'll see today—the briefing, the code, the verification evidence—was produced by one person using the workflow model we're proposing.
>
> The project is:
>
> - **Concrete enough to execute**—working code, real artifacts
> - **Abstract enough to remain vendor-neutral**—no specific tools required
> - **Self-demonstrating**—created using the workflow it proposes
>
> "

4.2 Problem Statement (10 minutes)

Walk through the eight constraints (Slide: Problem Statement):

1. Fragmented workflows across mobile, desktop, and execution environments
2. AI assistance unavailable or outside trust boundaries
3. Inbox-centric work with decisions buried in email

4. Untracked coordination limiting traceability
5. Knowledge attrition as personnel retire
6. Budget constraints reducing investment
7. Legacy systems facing decommissioning
8. Regulatory burden diverting resources

Emphasize: These are **systemic** constraints—not individual failures. The solution requires environmental change, not harder work.

4.3 Proposed Solution (10 minutes)

Walk through the workflow model (Slide: Proposed Workflow Model):

> "The workflow has three phases:
>
> **1. Ideation (Mobile):** Frame objectives, capture rough drafts, use AI to synthesize. This happens anywhere—on a phone, tablet, desktop.
>
> **2. Execution (IDE):** Turn drafts into versioned, reviewable artifacts. Build inside a development environment with AI assistance. Execute within trusted boundaries.
>
> **3. System of Record (Git):** Everything is captured automatically. History, rationale, decisions—all preserved for audit and knowledge transfer.
>
> The key insight is that **work flows freely between ideation and execution**, but **recording happens automatically**. No extra effort required."

4.4 Live Wireshark Demo (15 minutes)

This is the technical highlight. You'll demonstrate Bundle Protocol v7 actually on the wire.

Demo Setup:

1. Open Terminal
2. Navigate to repository: `cd ~/SpeakUp`
3. Have Wireshark installed (or use tshark)

Option A: Full Wireshark Demo (Recommended)

```
# Run the capture demo
./src/bpv7/capture_demo.sh
```

Explain while it runs:

> "What you're seeing is:
>
> 1. tcpdump capturing traffic on the loopback interface, port 4556
> 2. A BPv7 bundle being created and sent via TCPCL
> 3. The capture file opening in Wireshark

In Wireshark, I'll decode TCP port 4556 as TCPCL. You can see:

- The 'dtn!' magic bytes in the contact header
- CBOR-encoded bundle data
- The payload traversing a simulated DTN link

"

Option B: Console Demo (No Wireshark GUI)

```
# Run console-based test with hex dump
PYTHONPATH=src python3 src/bpv7/test_local.py
```

This shows:

- Bundle creation with CBOR encoding
- Hex dump of the bundle on the wire
- TCPCL contact header exchange
- Bundle reception and decoding

Option C: PDF Transfer with tshark Verification

```
# Transfer the briefing PDF via bundle protocol
PYTHONPATH=src python3 src/bpv7/test_pdf_transfer_tshark.py
```

Key points to highlight:

"This test sends the actual SpeakUp briefing PDF via Bundle Protocol. We're using tshark to capture and verify:

- TCPCL contact headers are on the wire ('dtn!' magic)
- CBOR-encoded bundle payload is visible
- MD5 integrity is verified end-to-end
- Standards compliance: RFC 9171 (BPv7), RFC 9174 (TCPCL), RFC 8949 (CBOR)

This proves the implementation actually works—not just unit tests, but **observable protocol behavior on the network**."

Wireshark Deep Dive (if time permits):

1. Open the saved .pcap file
2. Right-click TCP stream → Decode As → TCPCL
3. Filter: `tcpcl` or `bpv7`
4. Show the protocol hierarchy
5. Point out the 'dtn!' magic (hex: 64 74 6e 21)
6. Show CBOR array start (0x9F)

4.5 Verification Evidence (5 minutes)

Quick walkthrough of verification artifacts:

> "The project produces verification evidence as first-class artifacts. This isn't afterthought documentation—it's built into the workflow.
>
> **Automated Scans:**
>
> - PII scan—no sensitive personal data
> - Malware scan—ClamAV clean
> - Secrets scan—no API keys or credentials
> - MAC address scan—no hardware identifiers
>
> **Compliance:**
>
> - No CUI, no proprietary data, no classified information
> - NIST SP 800-53 control alignment documented
> - FIPS 199 security categorization: LOW
>
> The key point: **you can review this repository without relaxing any security rules**."

4.6 Value Proposition (5 minutes)

Emphasize the core message:

> "With the right environment, one person can do the work of an entire team.
>
> What you've seen today:
>
> - A complete BPv7 protocol implementation
> - TCP Convergence Layer for DTN communications
> - Professional briefing deck (LaTeX to PDF)
> - Verification and compliance evidence
> - Video production pipeline
> - Auditable cost tracking
>
> All produced by **one person in approximately 8 hours**.
>
> The constraint isn't capability. It's environment. If we provide the right tools, workflows, and trust boundaries, we can dramatically multiply productivity while maintaining—or improving—security and traceability."

4.7 Next Steps & Q&A (5 minutes)

Proposed next steps:

1. **Review the repository**—all source, documentation, and evidence is available
2. **Identify a pilot area**—where could this workflow be applied?

3. **Establish infrastructure**—repository, IDE access, AI tooling within boundaries
4. **Measure and iterate**—track outcomes, refine approach

Open for questions.

5. Demo Commands Quick Reference

Command	Description
`./src/bpv7/capture_demo.sh`	Full Wireshark capture demo
`PYTHONPATH=src python3 src/bpv7/test_local.py`	Console bundle test with hex dump
`PYTHONPATH=src python3 src/bpv7/test_pdf_transfer_tshark.py`	PDF transfer with tshark
`./build.sh`	Run full build (tests, scans, briefing, vide
`./verification/scripts/run-all-scans.sh`	Run security verification scans

6. Backup Materials

If questions arise about:

- **Standards:** Reference RFC 9171 (BPv7), RFC 9174 (TCPCL), RFC 8949 (CBOR)
- **Security:** Show `verification/Security-Attestation.md`
- **Costs:** Run `./accounting/calculate-costs.sh`
- **Architecture:** Walk through `src/bpv7/` directory structure
- **Requirements:** Show `verification/Requirements-Traceability.md`

This agenda was created using the SpeakUp workflow model it describes.

Repository: `https://github.com/brucedombrowski/SpeakUp`