# PRIME CONTRACTOR CAPABILITY BRIEF

**Bare-Metal Navigation & Cryptography Engine — Subcontractor Integration Opportunity**

DISTRIBUTION STATEMENT A
VERSION 2.0 · UNCLASSIFIED
ABRAMS RESEARCH LLC

> **From:** Abrams Research LLC (Loveland, Colorado)
> **To:** Lead Systems Engineers — Aerospace Prime Contractors
> **Subject:** Modular GPS-Denied PNT & Cryptographic State Record Engine for Legacy Flight Software Integration

## EXECUTIVE SUMMARY

We have built a bare-metal Rust navigation and state record engine designed to plug into existing C-based flight software without modifying the host system. We are seeking IRAD partnership or subcontract work to integrate this engine with proprietary ephemeris, sensor, and gravity datasets.

**What we built:** The engine block and the integration sockets.
**What we need from prime partners:** Target hardware integration and proprietary flight datasets to fill the sockets.

## 1. WHAT THE ENGINE DOES

The engine implements the Abrams Event Address (AEA) protocol — a 7-vector spacetime coordinate standard with cryptographic integrity sealing.

| Module | Function | Stack Size |
|---|---|---|
| `aea_7vector.rs` | 7-vector spacetime address with SHA-256 seal | 56 bytes |
| `ekf_filter.rs` | Extended Kalman Filter (6-state, 6×6 matrix) | 288 bytes |
| `rk4_propagator.rs` | RK4 N-body orbit propagator (Earth+Sun+Moon) | 48 bytes |
| `bcrs_frame.rs` | BCRS↔GCRS↔ITRS coordinate frame transforms | 72 bytes |
| `gps_denied_nav.rs` | Star tracker + gravimeter integration sockets | Stubs (todo!) |
| `cfs_adapter.rs` | C-ABI telemetry adapter | 70 + 42 bytes |
| `gf2_gauss.rs` | GF(2) Gaussian elimination (syndrome verification) | In-place |

**Total computation footprint:** < 500 bytes of stack. Zero heap allocation. Zero OS dependency. Single-threaded execution.

# PRIME CONTRACTOR CAPABILITY BRIEF

Section 2: C-ABI Integration Architecture



## 2. HOW IT INTEGRATES WITH YOUR FLIGHT SOFTWARE

The engine provides a packed C-ABI interface via `include/aea_cfs_msg.h`. Drop this header into your existing C build system:

```
// Drop this header into your existing build system
#include "aea_cfs_msg.h"

// 1. Construct a command packet
AeaComputeCmd cmd;               // Exactly 70 bytes, #pragma pack(1)
cmd.target_body = 399;           // Earth (Planetary NAIF ID)
cmd.epoch_sec   = 1893024000;    // TAI seconds since standard epoch
cmd.pos[0]      = 6778137.0;     // Cartesian Position X (meters)
cmd.pos[1]      = 0.0;
cmd.pos[2]      = 0.0;
cmd.vel[0]      = 0.0;           // Cartesian Velocity Vx (m/s)
cmd.vel[1]      = 7670.0;
cmd.vel[2]      = 0.0;

// 2. Call the bare-metal Rust engine
AeaResultTlm result;             // Exactly 42 bytes, #pragma pack(1)
aea_compute(&cmd, &result);

// 3. Inspect results
// result.status_code == 0 indicates SUCCESS
// result.morton_code contains compressed 8-byte 3D spatial index
// result.aea_hash contains SHA-256 cryptographic state seal
```

**Compilation:** Link against `libaea_core.a` (static library). No shared objects, no dynamic runtime dependencies.

**Struct sizes verified** via `sizeof()` on both GCC and the Rust compiler:

- `AeaComputeCmd`: exactly **70 bytes**
- `AeaResultTlm`: exactly **42 bytes**

# PRIME CONTRACTOR CAPABILITY BRIEF

Section 3 & 4: Sockets Architecture & Performance Audit



## 3. THE INTEGRATION SOCKETS — WHAT YOUR DATA FILLS

| Socket in Our Code | What It Needs | Your Resource |
|---|---|---|
| `bcrs_frame.rs` → precession matrix | IAU 2006 1,300-term polynomial | SOFA C library (public) or your internal equivalent |
| `rk4_propagator.rs` → body positions | Precise planetary positions | SPICE DE440 kernel (public) or your internal ephemeris |
| `gps_denied_nav.rs` → `lookup_star()` | Star catalog matching | Hipparcos/Tycho-2 (public) or your star tracker database |
| `gps_denied_nav.rs` → `lookup_geoid()` | Gravity anomaly grid | EGM2008 (NGA public) or your internal gravity model |
| Cross-compile target | HITL WCET measurement | SPARC LEON3 / RAD750 / ARM Cortex-R eval board |

Each socket is currently a Rust `todo!()` macro that panics at runtime if called — verified by `#[should_panic]` unit tests. This is deliberate: we built the architecture; host programs bring the mission data.

## 4. VERIFIED PERFORMANCE (INDEPENDENT AUDIT)

| Metric | Value | Measurement Context |
|---|---|---|
| RK4 orbit displacement | 177 km | Earth+Sun+Moon gravity, 1 LEO orbit propagation |
| Morton3D encoding | 8 bytes/coord | Lossless spatial coordinate roundtrip test |
| SHA-256 seal | < 0.001 ms | Determinism & tamper-detection verification |
| Unit test suite | 24 pass / 0 fail | `cargo test --release` on x86_64 host |
| Flight hardware WCET | Unmeasured | Pending HITL integration on target embedded processor |

# PRIME CONTRACTOR CAPABILITY BRIEF

Section 5 & 6: TRL Rating & Partnership Channels



## 5. HONEST TRL ASSESSMENT

| Component | TRL | Status & Verification Notes |
|---|---|---|
| SO(3) quaternion kinematics | 2–3 | Kinematic mathematics proven, no full 6-DOF dynamic simulation |
| AEA 7-Vector protocol | 2 | Architecture defined, packed C structs verified, SHA-256 seal operational |
| C-ABI Interface & Harness | 3 | Struct layouts matching GCC/Clang, static library linking validated |
| GPS-denied navigation sockets | 1 | Clean architectural sockets defined, awaiting sensor stream integration |

## 6. WHAT WE ARE ASKING FOR

**Option A — IRAD Subcontract:** We integrate our core engine into your flight software under your IRAD program. You provide the target testbed and datasets; we deliver the linked binary, Interface Control Document (ICD), and test harness.

**Option B — SBIR Subcontractor:** If your team is responding to upcoming DoD solicitations involving bare-metal navigation, GPS-denied PNT, or embedded state security, we partner as a specialized subcontractor providing the core algorithm engine.

**Option C — Independent Technical Evaluation:** We deliver the full source repository and C header package for your Lead Systems Engineering team to audit. Run `cargo test --release` locally to verify integrity.

## CONTACT & INSTITUTIONAL COORDINATES

**Joshua M. Abrams**, Principal Investigator
**Abrams Research LLC**
Loveland, Colorado, USA
Official Email: contact@abramsresearch.net
Portal: abramsresearch.net