Capabilities PDF (draft)

Title

Trading Infrastructure That Works

Tolga Berger — tolga@bergerandbergerglobal.com

One-liner

I build the systems trading teams rely on: exchange integrations, research/backtesting pipelines, execution tooling, microstructure analytics, and real-time data architecture.

Capabilities (with concrete technical detail)

Exchange integration (REST + WebSocket)

Built multi-venue adapters that normalize market data and order events, handle reconnect/backoff, and support order & position management primitives (place/replace/cancel, reduce-only, fills, risk snapshots). Designed for durable streaming (not "best effort" polling) so research and production share the same event model.

Backtesting infrastructure (proper methodology)

Implemented a repeatable experiment harness with parameterized strategy generation, train/validation/holdout splitting, and realistic cost modeling (fees, spread, slippage assumptions, funding). Output is structured and reproducible: same inputs → same reports, with explicit guardrails to prevent data leakage between splits.

Execution systems (safety + triggers)

Implemented on-exchange trigger orders with reduce-only flags and native OCO-style behavior (via reduce-only cleanup), eliminating poll-based stop execution that can introduce large excess slippage. Added order lifecycle safety: idempotent cancels, failure-mode handling, and kill-switch patterns.

Microstructure tooling (L2 + adverse selection)

Built L2 snapshot ingestion and analysis to quantify fill quality: queue depth, touch-through vs reversal rates, and adverse selection/toxicity measurement. This turns "looks good in sim" into a measurable live-deployment risk assessment.

Real-time data pipelines (ops included)

Designed WebSocket → durable logging pipelines (CSV/JSONL) deployed as persistent systemd services with auto-restart, rotation-friendly output, and replay capability. This makes the system operational: always-on collection, inspectable artifacts, and reliable backtests.

Engagement models

Architecture audit (1–2 weeks)

Review your current stack (data, research methodology, execution, ops). Deliver a prioritized roadmap with acceptance criteria.

Build sprint (4-6 weeks)

Ship one production subsystem end-to-end (integration / pipeline / execution / microstructure tooling) with tests + runbook.

Retainer (embedded support)

Ongoing infrastructure support: ship improvements, handle edge cases, keep systems reliable.

Contact

tolga@bergerandbergerglobal.com