Introducing Data Streaming System Interface for Twinscans

Openbare Samenvatting — KIVI EngD Award

ASML's Twinscan lithography systems generate massive volumes of diagnostic report data, which is currently processed through a batch-oriented pipeline that buffers data in memory and writes to the machine's local disk. As data volumes surge, this architecture faces severe CPU, memory, and disk I/O bottlenecks. To mitigate these risks and future-proof the infrastructure, this project introduces a highly scalable, real-time data streaming interface designed to forward data continuously as it is produced, completely bypassing local disk persistence.

The project was executed in two core phases: a rigorous benchmarking framework that evaluated six high-performance messaging technologies, followed by the development of a functional, containerized prototype. Based on the benchmark results, Apache Arrow Flight was selected as the optimal foundational architecture. The prototype successfully leverages Arrow Flight's zero-copy columnar data transport to establish a high-throughput streaming pipeline, empirically validating that the new interface can reliably broadcast telemetry while strictly complying with ASML's stringent hardware resource limitations.

> *"Yuhang took ownership of this challenging architectural study and quickly distinguished himself through his structured way of working and excellent communication skills. He systematically gathered requirements, assessed multiple technology options, and provided clear analyses of their respective strengths and trade-offs. The research results and functional prototype provide ASML with a valuable foundation and reference architecture for future product development. I am highly impressed by both the quality of his work and his professional growth."*

— **Flavius Olteanu**, Software Architect, ASML Netherlands B.V.

Challenges

Introducing a streaming interface into the highly complex Twinscan architecture required resolving substantial engineering ambiguity. A core challenge was sustaining high-volume data transmission while addressing the original system's severe CPU, memory, and disk footprint bottlenecks. Furthermore, the prototype had to operate within exceptionally strict resource constraints to avoid degrading the machine's core lithography performance.

Results

We rigorously benchmarked six messaging technologies, evaluating throughput, latency, and resource consumption. Apache Arrow Flight emerged as the optimal foundation due to its superior efficiency and compatible ecosystem. We subsequently developed a functional, containerized C++ prototype leveraging Arrow Flight's zero-copy columnar data transport.

Comprehensive testing validated that the prototype successfully broadcasts high-volume report data while strictly adhering to the mandated constraints.

Benefits

The validated prototype successfully demonstrates the feasibility of alleviating the resource bottlenecks within the original batch-and-persist pipeline. By bypassing local disk storage, it illustrates how a resource-heavy, disk-bound workload can be transformed into a highly scalable, network-bound pipeline. Ultimately, this project delivers a proven technical blueprint that enables real-time diagnostic capabilities and guides future product integration to handle exponentially escalating data volumes.

