
MIL.STRATML.SERVICES

a MultiPlex|studio production

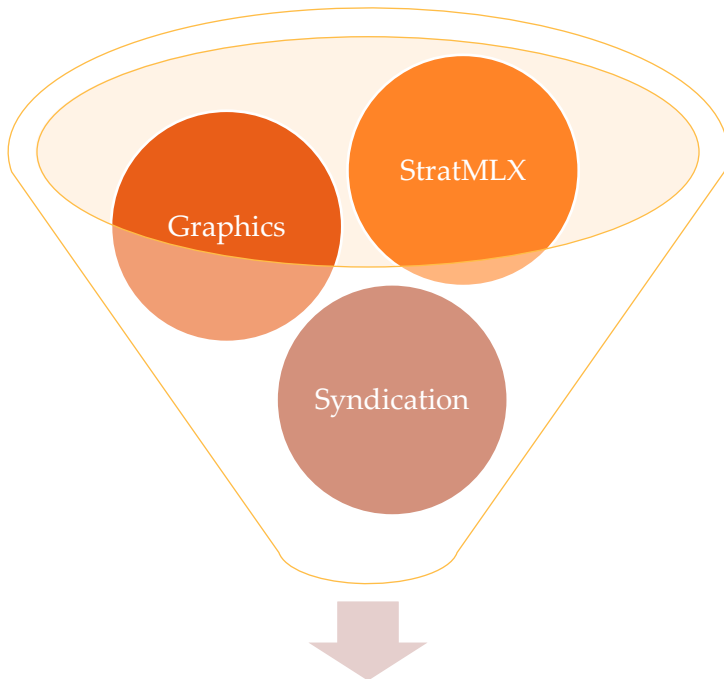
Jason L. Lind, USAF (Sep.)

Coalition Lead / lind@multiplex.studio

22 September 2020

A dire and accelerating need of the DoD, and really organizations in general, is to enforce their strategy downstream and verify upstream with reporting; StratML, an ISO (Part 1) and ANSI (Part 2) standard, begins to solve this problem and – with some extension, aka StratMLX – an architecture could be born to enable this next generation of C2.

StratMLX, whose proof-of-concept was implemented as StratML Part 3.5 in 2018, has several dozen elements in a consistent, yet complex and flexible, structure – and as such expecting anyone, particularly high level commanders, to develop the StratML by hand is a non-go thus requiring a Graphical Modeling Tool which is then syndicated through mil.StratML.services.



microServices

Existing C2 platforms are not going away any more than new C2 systems will be guaranteed to be on a common architecture, as such to achieve a vision of unified Command and Control we must consider a microservices based approach. Microservices enable connecting to a vast array of platforms and system for enforcement and reporting thus enhancing C2 across the DoD.

One area in particular this would be useful is energy as there is a concrete, yet abstract, “Flight Plan for Energy for 2017-2036” that defines the USAF strategy for achieving several goals and objectives relating to energy – particularly fuel consumption. A real world use case is to

set aircraft usage – flight plans, schedules and parameter (e.g. afterburners) – at the strategic (4*/3*) level and enforce that all the way to ground systems – programmatically – then reporting back compliance up to the top.

MIL.STRATML.SERVICES

a MultiPlex|studio production

Jason L. Lind, USAF (Sep.)
Coalition Lead / lind@multiplex.studio
21 September 2020

Strategy Markup Language is an ISO (Part 1) and ANSI (Part 2) standard for expressing strategic planning in machine readable format. While there has been a steady lack of development over the past 4 years on the standard it has not been abandoned and is the only markup language that contends to speak strategy.

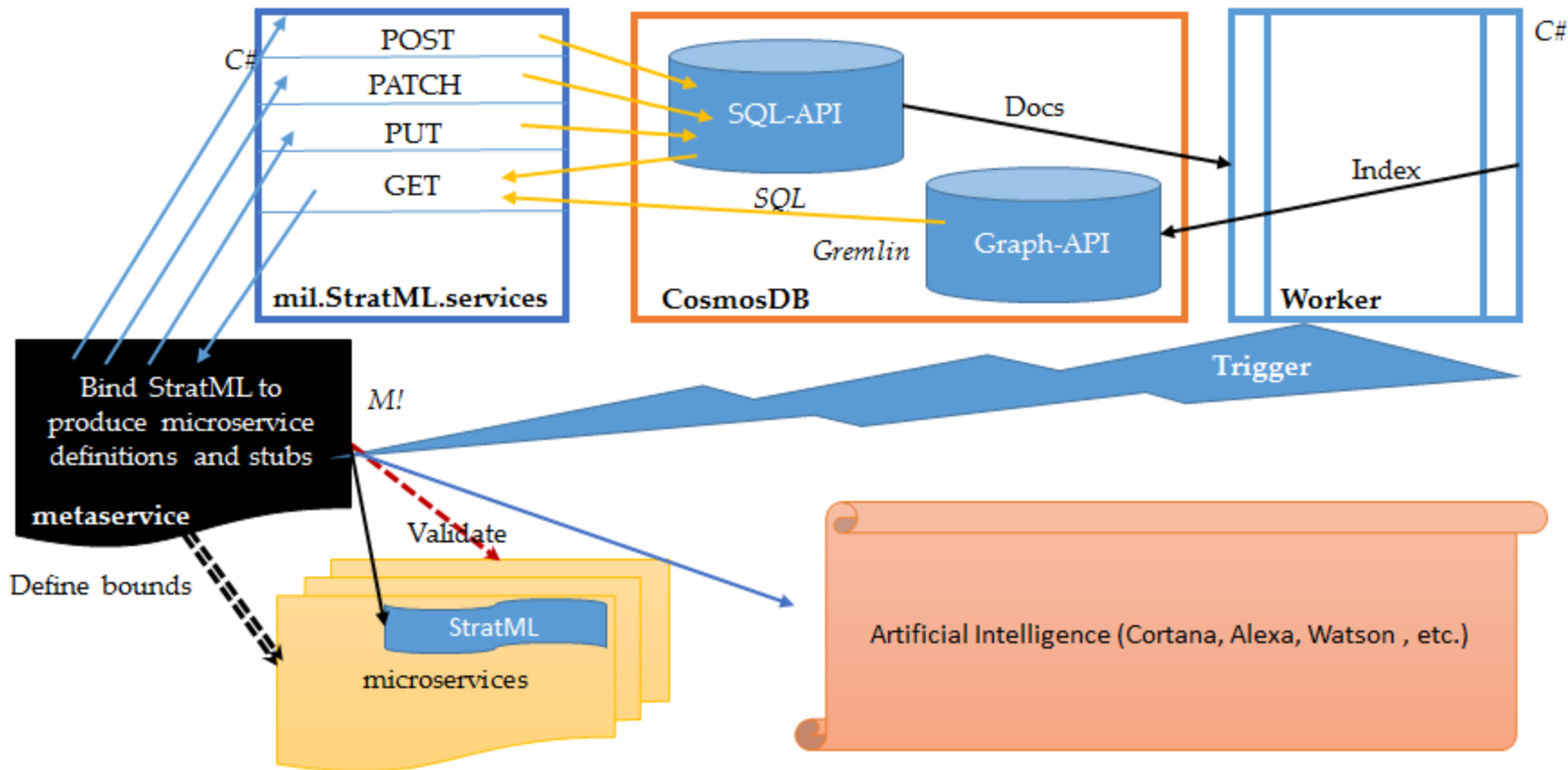
StratML.services seeks to build a syndicated API for storage, consumption, transformation and indexing of strategic plans. This is a greenfield development project that has a proof-of-concept (POC) code base from late 2018 published on github.com/jlind0.

While the civilian POC was based around consuming and indexing IRS Form 990 (disclosures for Non-Profits) the military one will be centered on perpetuating strategy decisions to Command-and-Control (C2) systems; specifically helping realize a DoD objective of a 20% reduction in aircraft fuel consumption, by 2022, via altering flight plans without any mechanical engineering improvements.

The mil.StratML.services project has several high level goals:

- Extension of StratML to part 4 which includes higher fidelity business logic and work with 3D PDF Consortium to develop it as an ANSI standard. While this will complicate the schema by several orders of magnitude (a part 3.5 was in the works in 2018) we contend that this lack of fidelity was one of the primary reasons for low adoption and support of the part 1 and 2 standards.
- Graphical Strategy Architecture (SA) tooling around StratML akin to Sparx Enterprise Architect, possibly even a plugin to existing UML toolkits.
- Syndicated services for distribution of, and analytics on, strategic plans.
- μ service and Composite Applications Catalog such that the community can develop μ services, and applications, that leverage the mil.StratML.services platform delivering C2 functionality on the near-endless ecosystems within the DoD.

While StratML is “markup” not “model” part 4 would be a combination of the two. We need to be able to model and distribute strategy plans in order for C2 to be enforced and reported up and down the chain of command. The initial use case will be exemplary of the power of this platform and should lead to rapid expansion of C2 functionality throughout not only DoD but Allied militaries as well.



The core of StratML.services is a No-SQL, in the implementation above via Azure CosmosDB, datastore that enables deep insights on StratML documents – both within, and more importantly across. The SQL-API of CosmosDB enables storing StratML documents in JSON – which can then be transformed via a worker process into Graph-API yielding deep-indexing. mil.StratML.services provides CRUQ (Create-Read-Update-Question) support to any number of 2nd and 3rd party services and applications.

The primary architecture implementation consideration is microservices, however for these to be useful they must react to changes in the plans without necessarily requiring a rewrite, and when the interface does change the stakeholders must be notified. A metaservice is written in the new M! (M-Bang) language and binds schema changes in StratML.services to a syndicated microservices catalog.