

Agenda - Hanoi Pre-wire Day 1

March 31

Attendance



Geneva Issues

- Race condition testing.
 - Actions
 - Mike & Iain – own task of looking at SDKs and associated services and make sure the makefiles include the race detector (-race command line flag for the Go tool set) in Go builds and tests
 - James indicate that there is a well known issue that race flag does not work with Alpine based images (as in CI/CD). Solving this issue is not in scope for Geneva. The work of instituting use of race flag in CI/CD must be postponed to Hanoi. See <https://github.com/golang/go/issues/14481>.
- Config seed PR – do we need mini-freeze for Lenny's PR?
 - Lenny keeping up with rebasing
 - Some bugs still being worked but nothing that requires mini-freeze at this time.

Architecture Tenet Discussion

Should EdgeX consider new tenets or architectural approaches given suggested needs of HA, more scalability and distribution support?

Statement about future project direction

The shape and size of “the edge” is changing. When initially conceived, EdgeX was designed and built to operate on resource constrained edge devices – such as edge gateways and small compute platforms like a Raspberry Pi (representative of the edge compute resource constraints).

Today, edge computing is expanding. Edge computing still addresses computing where the physical sensed world and operational technology (OT) meets the digital world and information technology (IT). While this was traditionally thought to be at the furthest extremes of computing resources, this is no longer the case. Now, OT environments collide with IT in environments

where compute resources are much more plentiful. In fact, the trend is to see edge and IoT computing expanding into use cases (exemplified by retail department store or smart factory environments) where compute resources are nearly equivalent to those in enterprise environments. The compute, network and storage availability in these environments support edge applications that can behave more like enterprise applications – that is be more distributed, more highly available, and scale more gracefully and would otherwise be indistinguishable from cloud-based applications.

To be sure, there are still use cases and edge application needs where compute, network and storage resource constraints require edge applications to remain relatively small and/or flexible to be distributed across the limited resources at the edge.

EdgeX remains committed, for now, to be able to be run on limited resource devices (Raspberry Pi in dimension) and environments (intermittent connectivity, shallow storage devices, etc.). It remains committed to its founding architectural tenets, optimal flexibility principals, and micro service architecture. Without endorsing new architectural tenets or principals at this time, EdgeX will however keep an eye on edge expansion and defining use cases. Depending on user requirements and the evolution of edge compute, EdgeX may embrace the needs of higher order compute environments in the future and lean toward

- Removing services (in favor of technology provided infrastructure – for example removing a logging service in favor of a central logging facility provided by the deployment/orchestration technology)
- Collapsing services (combining services that would not ordinarily be separate micro services in a more resource rich environment – collapsing services to save on complexity and service communication overhead)
- Instituting high availability design into services (such as 12 factor application design principles into services to support HA infrastructure and tooling)

The degree to which the project moves in this direction will be reviewed with each release planning session and subject to user requirements dictating any move in this direction.

If we can align on these basic questions, there is a feeling that this should start to guide our opinion on scope for Hanoi and even beyond.

Hanoi planning

General / Cross Cutting Concern

- Probably going to be a 1.3 vs 2.0 release
- Ability to institute all of the V2 API and some of the desired non-backward compatible features during one 6-month cycle is dubious.

New feature – tentatively included in Hanoi release

- Move to Go 1.15
 - Anthony Bonafide to do research and provide impact statement (also consider 1.14)

- Implement a message bus between device services and application services. As a stretch goal, core data would be an optional subscriber on the bus (persisting Events/Reading found on the bus); otherwise this feature would assume persistence is off
 - Lenny to present high-level design at the planning meeting (including recommendation on broker vs brokerless vs option for both)
- Allow device services to be distributed to alternate hosts.
 - Must address questions of security and orchestration
 - Must address device service to core as well as command to device service communications
 - All TSC members are requested to be familiar with Bryon's Option B regarding security
 - Bryon/Tony to send out slide deck or document to TSC for review that refines the Option B proposal and looks specifically at DS distribution question.
- As Hanoi is looking less like a version 2.0, should we reconsider LTS? Should the next 1.x release be LTS?

Tech Debt - tentatively included in Hanoi release

- V2 API implemented as "experimental" or alternate API while V1 is deprecated but still available
 - IOTech to provide readout on timing and considerations
- Deprecate Mongo
 - Requires Redis with secure access to be on first (Jim checking with Andre)
 - If so, mark Mongo as deprecated for Geneva and explore ridding project of Mongo for Hanoi.
- Remove value descriptor
 - For backward compatibility, can only be removed if in a 2.0 situation
- Devices with embedded device profile and removing addressable from EdgeX
 - For backward compatibility, can only be removed if in a 2.0 situation
- Develop process for security and license vetting of 3rd party components
 - Security WG (Tingyu and Diana with WG inputs) to provide a high level idea of the process and what it would require of developers and work groups
 - License compatibility with project also to be considered part of this review
- Getting consistent and legal with license and attribution files/marks
 - This topic is schedule for upcoming architect's meeting and will be reviewed at planning meeting if not sufficiently answered
- Restructuring our compose files to take advantage of compose file overrides, which removed the duplication in all our compose files. See <https://devilbox.readthedocs.io/en/latest/configuration-files/docker-compose-override-yml.html>
 - Lenny to provide readout/demo for consideration at the planning meeting

Core / Supporting

Tech Debt

- Tests should run cleanly when passed the "-race" argument
 - Will require DevOps assistance
- Restructure the Redis DBClient implementation for Event/Reading

- Normal refactoring exercise – does not require consideration at the planning meeting but will be looked into
- <https://github.com/edgexfoundry/edgex-go/issues/2166>