HR001120S0058
Verified Security and Performance Enhancement of Large Legacy Software (V-SPELLS)
Frequently Asked Questions

As of August 3, 2020

Q63: The contracting presentation said that combined proposals must address all TA1, TA2, and TA3. Sergey said any combination of the three.

A63: All combinations of TA1, TA2, and TA3 are allowed per the BAA. Each single TA proposal is limited to 30 pages, proposals that combine two to three TAs are limited to 50 pages. See BAA HR001120S0058 Amendment 1.

Q62: Is there a way of getting access to any documentation or recordings of the event and does not attending the proposers day event preclude a vendor from proposing on this acquisition? Also, has DARPA determined an acquisition strategy for this solicitation yet? BAA, OTA or other?

A62: You are welcome to submit to this opportunity, there was not a requirement to attend the proposer's day. Please see the following BAA link:

https://beta.sam.gov/opp/7dc5798bf5e74d8aa3df767edd3e0815/view#general

. There will be updates posted including answers to FAQ's and slides from the proposer's day. All documents associated with the opportunities are posted to the DARPA opportunities page at the following link: http://www.darpa.mil/work-with-us/opportunities?tFilter=&oFilter=3&sort=date

Q61: Where is the BAA posted?

A61: BAA is posted at the following link:

https://beta.sam.gov/opp/7dc5798bf5e74d8aa3df767edd3e0815/view#general

Q60: Please provide details on the item missing from the original V-SPELLS BAA (deliverables) that was included in the updated Addendum

A60: Phased and Final Technical Reporting – Phase 1 and Phase 2 reports are due at the end of their respective phase. The reports, due at phased or contract completion, will concisely summarize the effort conducted and provide any lessons learned during the development of the V-SPELLS technology.

Q59: Are sub-contractors permitted to participate in multiple proposals (excluding for TA4)?

A59: Yes.

Q58: Is there any constraint on DSL such as open source, freeware or in-house DSLs? Can the iterative process go from TA1 to TA2 to TA3 and come back to TA1 to be repeated again?
A58: The BAA encourages open source technologies but is not prescriptive with respect to particular DSLs to be used or created. Close and iterative collaboration among TA1, TA2, and TA3 is envisioned; see also A44.

Q57: Might it be the case that multiple DSLs arise from a single target system in order to support composition of different aspects of the system? For example, lifting into one DSL to control the network behavior of a component while lifting a separate DSL to control physical pumps over General Purpose Input Output (GPIO)?
A57: Yes. Safe composition of multiple DSLs is envisioned among the program challenges.

Q56: Is hardware provided if the program involves hardware?

A56: The TA4 performer is expected to curate and provide hardware or access to hardware for evaluation when necessary. Please also see A32.

Q55: Can you elaborate on the "automated, iterative interactive" aspects of TA1 technology—automated and interactive are kind of at odds with each other and could you explain their interplay/level of acceptable manual interaction?

A55: It is expected that substantial and novel automation will enable human domain-expert developers engaged in replacement or enhancement of legacy systems to achieve the goals of the BAA. The BAA recognizes that practical program understanding requires and starts from domain expert knowledge, and that automation leverages this knowledge interactively and iteratively.

Q54: Is the legacy code base provided with instructions on how to compile and run? Is there a case only binary provided without source code?

A54: Yes. Please see A26 and A27, and also A23.

Q53: Is the choice of domains for DSL's targeted by the program driven by expertise from TA1-3, the TA4 integrator, or the transition targets?

A53: Please see A20.

Q52: Is development of next-generation ABI's predominately the providence of TA3 or is it a joint effort of TA1-3?

A52: It is expected that TA3 will lead this development, in close collaboration with TA2.

Q51: How do we collaborate between TA's, for instance TA1 and TA2? How do we know what DSL TA1 uses to start the work on TA2?

A51: Please see A38 and A18.

Q50: To what extent is it desired for hardware models to be incorporated into the operational semantics of extracted DSLs?

A50: Strong proposals would consider exploration of hardware interfaces to validate their models.

Q49: Are the mix of different programming languages expected?

A49: Please see A35.

Q48: As an add-on was DevSecOps looked at as being an insufficient process for this DoD legacy code issue and would it be considered a starting point or is this considered a completely different task area?

A48: The BAA makes no such implication. Please see A43.

Q47: What does it mean by virtual machine extracted from low-level operations?

A47: Virtual machine is a technical metaphor for domain-specific operational semantics.

Q46: Do you anticipate that every legacy program that goes through the V-Spells tool chain may give rise to a new DSL, capturing the "domain" of that program? Or would a smaller number of DSLs applicable to wider domains be appropriate?

A46: The appropriate number of DSLs depends on the proposed solution and how well it achieves the goals of the V-SPELLS program. DSLs suitable for categories of legacy systems would strengthen the proposal if they increase its practicality.

Q45: One of the metrics highlighted is "initial memory load reduction". What does this refer to?
A45: Initial memory load reduction refers to the size of the executable code in the memory footprint.

Q44: Are there interactive iterative processes between TAs or internal to each TA?
A44: Both are envisioned and addressed in the BAA.

Q43: Much of this program seems to incorporate the principles of DevSecOps as outlined by the DoD. How does this differ from that?
A43: The program seeks to provide theoretical foundation and practical tools for effective maintenance and enhancement of legacy software. As such, the program shares some goals with DevSecOps and aims to contribute to these goals.

Q42: How will the correctness of TA1 produced DSL code be measured? Proof of equivalence to original code? Testing?
A42: Strong TA1 proposals would aim to provide the strongest assurance possible for legacy code bases.

Q41: What about conflicts between correctness and compatibility? (i.e., when there was a bug in the original code, the derived DSL (after improvement to ensure correctness) does not have that bug and the corrected behavior is incompatible with another module that was (implicitly) expecting the incorrect behavior?)
A41: It is expected that practical solutions would consider allowing the developer to account for bug compatibility where it matters to the BAA goals of assured replacement or enhancement of legacy components.

Q40: Does this require formal methods-based enhancement of genetic programming, eg. constraints on what mutations and recombinations are allowed because they preserve correctness and/or compatibility?
A40: See A39.

Q39: With regards to the technical definition of "construction": Can "correct by construction" and/or "compatible by construction" be addressed using evolutionary computation as a means of construction, as in the "genetic improvement" software engineering (testing, bug-fixing, performance optimization, etc.) application of genetic programming?
A39: Any means of construction for which resulting assurance guarantees can be verified are in scope.

Q38: Is the definition (syntax, semantics, and verification systems) of the DSLs part of TA1 or TA2? The BAA says "TA1's analysis tools are required to provide the extracted domain model and domain-tuned structures, as well as, the architectural information recovered from the legacy code base, to TA2's reasoning about the component specifications and DSL code enhancements, as well as, for TA2's compatibility and compositional safety analysis of the new DSL code" which suggests that the actual DSL definition comes from TA2.
A38: TA1 and TA2 are expected to closely collaborate on defining, deriving, and inferring the appropriate DSLs and semantic models to achieve the goals of the respective TAs, such as automated, interactive, iterative program understanding and compositional DSL programming.

Q37: Can you give examples of "domain virtual machines?"
A37: This term is used to mean well-structured implementations of operational semantics for domain-specific data types and structures.

Q36: Are you envisioning a pragmatic DSL (e.g., Java-based) or a formal DSL (e.g., Algebra-based) ?
A36: The BAA is not prescriptive in regard to DSLs and approaches to be used, but emphasizes relevance and effectiveness for large legacy code bases. The BAA further emphasizes achieving the strongest assurance guarantees possible for legacy code bases.

Q35: Although the "legacy code" term is understood intuitively, is it expected/advised to focus on specific programming languages?
A35: Strong proposals would consider languages relevant to DoD legacy code bases. Addressing C/C++ is envisioned.

Q34: Could you clarify what you mean by "hook system"?
A34: This term is used to refer to designs and mechanisms for composing newly developed code with a pre-existing system. The BAA emphasizes safety of such composition.

Q33: Do you expect to see formal proofs of correctness for the TA3 generated components?
A33: Strong proposals would consider providing the strongest assurance guarantees possible for legacy systems in practice.

Q32: For Evaluation platforms and test beds, will TA4 be responsible for furnishing hardware to the rest of the team or will DAPRA provide assistance?
A32: Strong TA4 proposals would consider covering the full scope of evaluation activities, including curating and providing hardware where necessary.

Q31: How close should the DSL abstraction be to the implementation of the legacy software? Is a behavioral abstraction reasonable?
A31: The BAA emphasizes effectiveness of practical component replacement or enhancement in a large code base. It is not prescriptive of particular abstractions to be leveraged.

Q30: How generic must the DSL extraction tool be with respect to domains? Is it acceptable for the tool to support creating DSLs only for a set of domains chosen a priori? E.g., would a tool be acceptable if it is very good at extracting a "networking" DSL from C code that manipulates network packet fields, but not good at extracting a DSL from arbitrary C code. Or, must the tool strictly extract a full DSL from any code?
A30: Strong proposals would consider methods that apply across multiple domains of interest, including those mentioned in the BAA and are practical for large code bases. It is, however, understood that a single universal methodology suitable for arbitrary C code may not be possible.

Q29: Is it reasonable to expect that the system will be runnable and/or that test cases (or inputs) would be available?
A29: Yes. However, performers should not assume that any available tests will be exhaustive or functionally complete.

Q28: Can foreign research institutions participate as subcontractors? Will TA4 evaluation (and possible inclusion of DoD systems) preclude foreign organizations from participating in the program?
A28: Foreign researchers may participate, subject to relevant laws and regulations. Please refer to the Contract Management Office briefing slides. Evaluation challenges prepared by TA4 for TA1, TA2, and TA3 performers will include no Controlled Technical Information (CTI). However, TA4 performer will be expected to have suitable access to DoD systems to facilitate transition, and will include appropriately cleared personnel, as described in the BAA.

Q27: Is it correct to assume that for legacy code, tool settings (i.e. compiler settings, etc.) are known? Will sample legacy code and settings be provided to performers?

A27: Proposers may assume that the build process is available. Evaluation challenges will provide the code and build process.

Q26: Can TA1 expect to have compilable source code and/or runnable binaries for analysis? Is TA1 expected to be a strictly static analysis, or is the use of dynamic analysis in scope?
A26: Yes, proposers may assume that compilable source code is available, with possible exceptions of small well-specific enclaves that are available for interaction. Dynamic analysis is in scope.

Q25: For TA3 packaging proofs with ABIs, does a security consideration come into play? I can imagine that it is possible to attach proofs with linkers, but these may be easily hacked and tampered with. So, would we have to devise strategies such that the proofs cannot be hacked?
A25: Strong proposals would consider practical security considerations for the novel ABI extensions they develop.

Q24: The BAA references cyber-physical systems, for such systems some constraints (such as an operating safety envelope) may not be directly reflected in software. Should solutions attempt to infer such constraints, or is this expected to be contributed directly through interaction with domain experts?
A24: Strong proposals would consider both of these cases.

Q23: Would it be appropriate to consider legacy software settings for TA1 where sources (or components thereof) are not available?
A23: Yes. However, proposers may assume that source code is predominantly available.

Q22: How much human assistance (from the developer) is acceptable for the DSL extraction tool? For example, an application that both manipulates network packets and draws shapes on a screen, may warrant extracting two different DSLs from respective parts of the codebase: Is it acceptable for the human to decide the subsets of the code from which to generate DSLs?
A22: Strong proposals would consider practical aspects of analyzing legacy source code bases. Some human expert interactive participation is expected.

Q21: Should we address the aspects related to multicore platforms and related issues like dynamic allocation of the memory, shared resources between CPU, etc.?
A21: All of these considerations are in scope.

Q20: Do you envision the choice of domains for DSLs developed by TA1 (and supported by TA2-3) driven by proposals from TA1-3, by the TA4 evaluator, or by transition use cases?
A20: Strong proposals would consider techniques that will apply across multiple domains, including those mentioned in the BAA. Successful TA1, TA2, and TA3 performers will be able to effectively address challenges curated by TA4 and their performance will be evaluated on these challenges. Strong proposals would aim for effective transition.

Q19: Would you clarify the word "domain", this is very broad. Are there specific ones that we should focus on?
A19: The BAA uses the word “domain” in the sense that it is commonly used in programming languages research when referring to “domain specific languages”. For example, it is assumed that there exist sets of non-trivial data types and structures associated with the domain and sets of well-defined operations on these types, which can be described formally and unambiguously. The BAA mentions several examples of domains of interest.

Q18: I understood that TA2 will be performing program analysis over code written in DSLs recovered by TA1, where the DSL is presumably ultimately specified by some sort of grammar and

operational semantics, and in turn, that DSL grammar and operational semantics will itself assumedly be recorded in an agreed upon DSL specification language. Will all the TA1 performers be expected to recover DSLs in the same DSL specification format, so that all TA2 solutions are compatible with all TA1 solutions? And if so, who will be responsible for creating that cross-TA interchange language for defining recovered DSLs?
A18: TA1 performers are expected to closely collaborate with TA2 performers on the means and methods of iterative program understanding, including the construction of DSLs and semantic models. The responsibility for creating the means of relevant information interchange will be shared, and strong proposals would include technical plans for collaboration between TAs.

Q17: Does the DSL extracted (for TA1) need to be complete in some sense, or does it only need to be sufficient to allow the existing legacy software to be programmed?
A17: The DSL need not conform to any theoretical idea of completeness, as long as it is sufficient to address the BAA goals, such as safe composable enhancement.

Q16: For TA3, are the following approaches in scope? Middle-wares, approaches requiring recompilation after a deployment change (configuration files a-la Oil for Autosar)? Or the scope is strictly ABIs and dynamic linking/loading?
A16: All approaches relevant to the BAA goals are in scope.

Q15: Can it be safely assumed that what is called here "legacy code" is code that does not carry along the source-language (compiled) compiler as a runtime-available function for on-the- fly synthesized code compilation?
A15: Yes.

Q14: Is deploying the code on hardware in scope? If so, is there a set of hardware the proposers should aim for? Is it sufficient to have the code built for different OSs or should we target multiple types of hardware such as GPUs, ASICS, etc.?
A14: Hardware-supported approaches are in scope, however, TA1-TA3 performers are not expected to provide any custom hardware as a part of their approach.

Q13: From my understanding, DSLs are automatically derived from legacy code. Is this expected to be a supervised, semi-supervised, or unsupervised process? Are DSLs expected to be unique to each legacy system or can/should they target categories of legacy systems?
A13: Strong proposals would consider practical aspects of analyzing legacy source code bases. Some interactive participation from human experts is expected. DSLs suitable for categories of legacy systems would strengthen the proposal if they increase its practicality.

Q12: Are there specific languages we should expect to be able to handle for the legacy code? Are there languages that are excluded for legacy code?
A12: Strong proposals would consider languages relevant to the DoD legacy code bases. Addressing C/C++ is envisioned.

Q11: Are security enhancements to legacy code also in scope? E.g., hardening software code against hardware attacks such as fault injection?
A11: Yes, it is in scope under the goals of the BAA. However, strong proposals would aim to address the broad range of enhancements outlined in the BAA, rather than focus on security enhancements narrowly targeting any particular threat.

Q10: Does TA1 need to take a language-agnostic approach? Should performers select the languages their system will work with? Are any specific source languages prioritized?

A10: Strong proposals would consider languages relevant to the DoD legacy code bases. Addressing C/C++ is envisioned.

Q9: Are there specific DSLs the program is targeting for?
A9: The BAA is not prescriptive with respect to specific DSLs or DSL technologies.

Q8: In terms of legacy code targeted, (1) should we be targeting both source code and binaries? (2) Are we free to pick our internal legacy code as a use case to demonstrate the system?
A8: Proposers may presume that the source code is available with the exception of small opaque enclaves that are well-specified and available for interaction. The BAA emphasizes the use of open code source bases to demonstrate the effectiveness of the proposed approach but any additional discussion of internal legacy code is allowed.

Q7: Could you provide a couple of examples of the "successful industry DSL s" that Sergey mentioned in his presentation?
A7: The BAA and the slides provide examples of the domains where DSL successes were achieved. No specific technology is prescribed by the BAA. The V-SPELLS slides and FAQ will be on the DARPA Opportunities Page.

Q6: Is it envisioned that the performers will need to produce verified code? Or, is it sufficient to produce verified composition and performance/security? As alluded to, generating verifiable code for legacy software may be impossible due to the absence of specifications.
A6: Strong proposals would aim to provide the strongest assurance possible for legacy code bases.

Q5: Sergey mentioned side channel protection as part of his presentation. Under what TA should this be? Also, I'm assuming that we are mining for automated protection as opposed to manual engineering. Is this correct?
A5: The BAA makes no specific emphasis on side channel protection, but strong proposals may want to address enhancement scenarios in which parts of the software are distributed to separate computing nodes or hardware enclaves, among other scenarios described in the BAA.

Q4: Will the existing software from which TA1 will be extracting DSLs from be in some specific programming language (e.g., Ada) or in several languages (e.g, COBOL Ada, C+ +)?
A4: Strong proposals would consider languages representative of the DoD code base. Addressing C/C++ is envisioned. Strong TA1 proposals would aim to provide the strongest assurance possible for legacy code bases.

Q3: How much of TA3 is expected to be formal? Do you envision that TA1 and TA2 produce coq theories and TA3 will be guided by those theories?
A3: Strong TA3 proposals would consider technologies that offer strong assurance guarantees. The BAA is not prescriptive with respect to any particular theories or tools. Strong proposals would aim to provide the strongest assurance possible for legacy code bases.

Q2: The metrics seem to be for the program. Are there specific metric for each TA?
A2: The metrics are all outlined in the BAA's Table 1. Strong proposals would offer additional metrics, specific to the TA(s) that they address.

Q1: Are software construction techniques such as Evolutionary Programming in scope?
A1: All software construction methods are in scope, so long as they provide strong assurance guarantees.