

Project Proposal to MExE for a Mobile Services Framework based on Common Language Infrastructure

**Mike D. Smith
Group Program Manager
Developer Division
Microsoft Corporation**

Microsoft[®]

Agenda

- **Background / Context**
- **Mobile Frameworks Proposal**



**Common Language
Infrastructure
Standardization Project**

Brief History

- **Common Language Infrastructure (CLI) is the specification for a multi-language runtime environment designed to fit many devices and platforms**
- **Proposed to ECMA TC39 for standardization**
- **Complements and support the work that is already ongoing in TC39**
 - **ECMAScript**

Status of Proposal

- **July 2000**
 - Initial proposal made to ECMA TC39
 - Strong interest evidenced
- **Sept 28 2000**
 - Next TC39 meeting
 - Proposal to add to TC39 work programme
- **If approved...**
 - Work most likely to begin later in 2000

An Open Language Model

- CLI is language neutral
 - All languages can be first class players
 - Leverage existing development skills
 - Leverage existing code
 - Future proofing
- Extensible language support
 - ECMAScript, C/C++, C#, VB, ...
 - WMLScript?
 - APL, COBOL, Eiffel, Haskell, ML, Oberon, Pascal, Perl, Python, Scheme, Smalltalk, ...
(many research & industry language partners)

A Profiled Specification

- **CLI is a profiled specification**
 - **Full-featured profile for desktop/server-level devices**
 - **Economy-focused profile for embedded devices**
 - **Open and flexible**
 - **“Vertical” frameworks for specific scenario categories layer on top**

General Design Considerations

- **CLI design considers the following:**
 - **Scalable Security central to the design**
 - **Economy / Performance tradeoffs**
 - **Adaptability / Profiles**
 - **Application and tools compatibility**
 - **Portability and easy device integration**
 - **Seamless connectivity**
 - **Peaceful real-time coexistence**

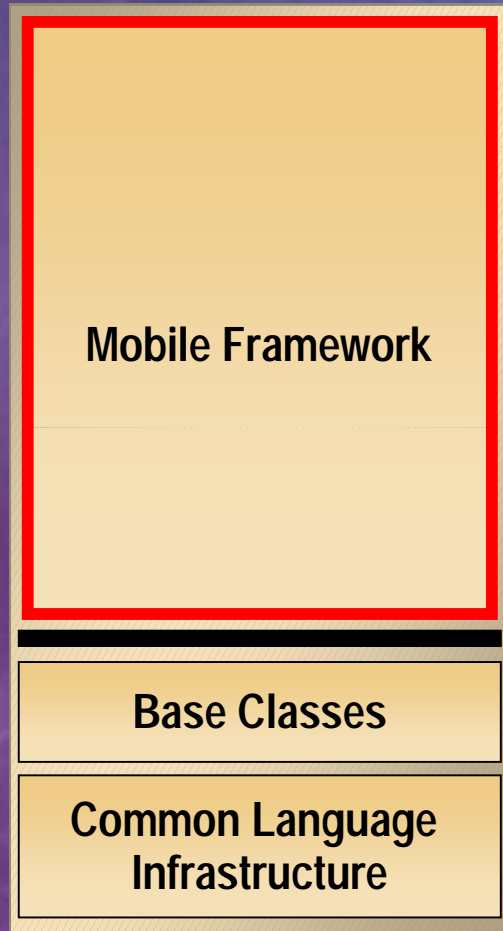
Mobile Considerations

- **Economy**
 - Efficient battery use
 - RAM: ~ 128 KB
 - Flash / ROM: ~ 512 KB
- **Good performance**
 - JIT compilation
 - Ability to trade off performance against resource usage (RAM, Flash, ROM)
- **Granular and configurable security model**
- **XML-based transport-independent protocols enable provisioning, cross-device interop, web services**

Mobile Frameworks Proposal

Proposal to MExE

Work together to define and standardize a CLI-based Mobile Services programming framework.



1. Collaborate on Mobile Framework classes



2. Leverage CLI base profile as open app execution environment. Ensure suitable profile, policies, configuration.

Potential Benefits

- **Broad developer accessibility**
- **New application scenarios**
- **Granular security model**
- **Cost-effective solution**
- **Synergy with existing MExE work**
- **Optional - support for ECMAScript**

Broad Developer Accessibility

- **Multi-language support**
 - All languages can be first class players
 - Leverage existing development skills
 - Leverage existing code
 - Future proofing the platform
- **Reach the full developer community**
 - ~ 6 MM professional devs worldwide
- **Broad partner support**
 - Tools vendors and language researchers
 - Many choices for developers

Granular Security Model

- **Scaleable security model – developer consistency**
 - Untrusted
 - Trusted domains (operator, manufacturer..)
 - Rich, flexible permissions & policies
- **Code access security**
- **Type safe and verifiable code**

Cost-Effective Design

- **Profiled standard**
 - Targets realistic hardware constraints
 - Scales up and down
- **Trade off functionality and runtime performance against use of battery power, RAM, Flash and ROM**

Synergy with MExE Work

- **Potential for synergy with CM2, CM3**
 - Hardware constraints
 - Reference points for multi-language mobile programming framework
 - Potential for API interoperability
- **Potential for optional integration/interoperability with Classmark 1**
 - Rich programming framework
 - Domain based security for code actions
 - Higher performance for MExE applications and Script

Optional Support for ECMAScript

- Potential synergy with WAP-NG investigation of ECMAScript
- ECMAScript as a compiled language
 - Higher performance
 - Code access security
 - Availability of programming framework

Looking Ahead

- **We believe the proposed project has good potential to enhance and complement MExE's mission**
- **We welcome your feedback**
- **We look forward to working with you on this opportunity**