Cincom®

Protocols: Past, Present & Future

By Jerry Kott
@cincomsmalltalk (Twitter)
#ESUG16

Protocols - the team

Develops networking, security, and web application development tools

Has a continuous history of about 20 years at Cincom Smalltalk

Exciting

Constantly evolving

Challenging

Innovative

Examples:

In ~ 20 years, we've had:

- **SSLv1**, **SSLv2**, **SSLv3**
- **TLSv1.0**, **TLSv1.1**, **TLSv1.2**
- **TLSv1.3** is a working draft

On the other hand:

- **HTTP/1.1** since 1997, but 8 different RFCs, some obsoleting others
- **HTTP/2** spec finalized in May 2015

Evolution at different speeds

What’s special about Protocols?

Mostly well defined - in contrast with many end-user application requirements

Very large audience (e.g.: **IP**, **TCP**, **HTTP**, **TLS** - practically every computer and every user in the world)

Small bugs can have potentially disastrous consequences (e.g., SSL buffer overruns and other security vulnerabilities)

Fast reaction is required (e.g., a server upgrades cipher suites, and suddenly Smalltalk client code breaks)

AppeX

A Web Development **Framework** and **Toolset**

HTML5 and ECMAScript 5.0 (a.k.a. **JavaScript**)

Introduced in 2014 as part of Cincom® VisualWorks® 8.0

https://www.youtube.com/watch?v=YmQXVnB9BO4

Added support for back-end database (**Glorp** & **ActiveRecord**) in VisualWorks 8.1 in 2015

Is there anything else we could do?

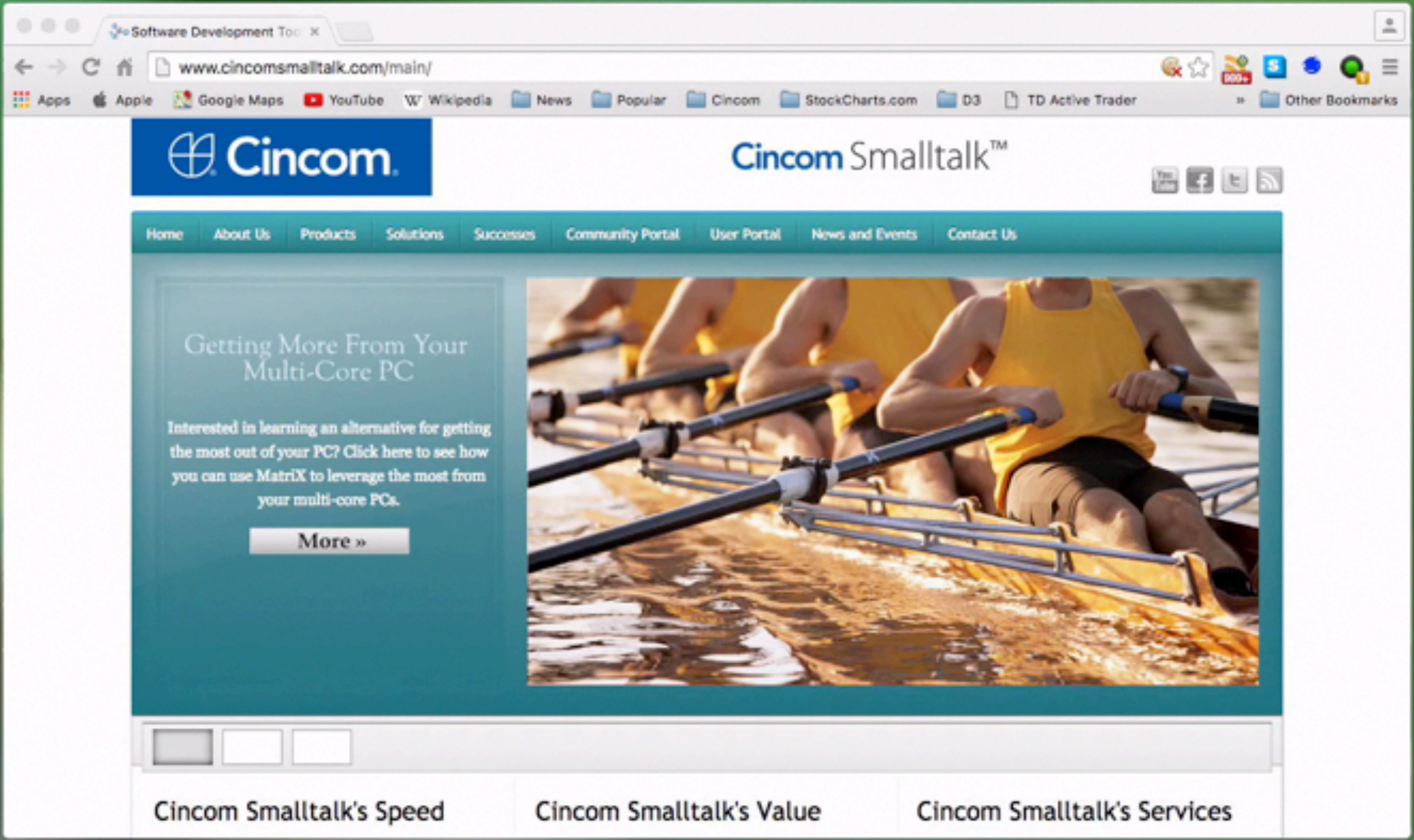
YES!

AppeX - improvements

AppeX-Scaffolding

- Included as a ‘preview’ component in VW 8.1
- Fully integrated in the upcoming release of VW 8.2
- Added a new GUI tool to generate a web application from an existing database.

Scaffolding Screen Cast



Cincom Smalltalk™

Getting More From Your Multi-Core PC

Interested in learning an alternative for getting the most out of your PC? Click here to see how you can use MatriX to leverage the most from your multi-core PCs.

Cincom Smalltalk's Speed

Cincom Smalltalk's Value

Cincom Smalltalk's Services

JavaScript Minification

Some of Single Page Application goals:

- Low Bandwidth Demands
- Obfuscate code for deployment

JS code gets stripped of comments and white space

Variable names get shortened and scrambled

Code size reduction dependent on style and naming conventions, range between 40% to 70%

AppeX-Minification in the AppeX-Tools parcel.

Minification Example

ApplicationClient class (AppeX-Core.js)

```
ApplicationClient javascript inspect
```

To turn minification on or off, use ‘Web Development’ submenu of ‘Tools’:

- Cache Minified Code
- Reset Minified Code Cache

Code size

- before: ~30 kB
- after: ~10 kB
- reduction of ~65 %

Shared Session Support

Allows to share session information among different AppeX application responders.

Example:

http://localhost:8889/shared-session

Different ‘pages’ of the /shared-session site have the same AppeX sessionId in ‘sessionStorage’.

On the server, the responders for each of the application classes share a single instance of SessionCache

3rd party JavaScript code

Not Cincom, not yours

For example: jQuery

```
<script src="//code.jquery.com/jquery-2.2.1.min.js" …
```

- No need to remember, just use a shared variable: **Application.JQueryLib**
- Easy to update as jQuery versions change (no hard-coded values)
- Predefined values for jQuery and D3
- More can be added (e.g., angular.js, bootstrap.js, etc…)

AppeX Internationalization

Use existing UserMessage framework within VisualWorks

Load **AppeX-Internationalization** parcel to add the required extensions.

A localized 'Hello World' example is included in the **AppeX-Examples** parcel

http://localhost:8889/hello-localized

UserMessage catalogs can be generated from a context menu of an Application class

e.g.: HelloLocalized popup menu -> 'Generate Catalog'

JSFile

Round-trip JavaScript Development

Synchronizes AppeX source code with a cached copy in the local file system

Currently working with Chrome Developer Tools

Maps a network resource to a local file

e.g.: http://localhost:8889/AppeX.CoreCode.js maps to file:///…/AppeX.CoreCode.js

Debug in web browser Developer Tools, save the changes

The modified code is automatically synchronized in VisualWorks IDE

A full demo included in my presentation on Friday at 10:00am.

HTTP/2

Faster, more efficient than HTTP/1.1

Multiplexing concurrent requests on a single TCP connection

Included as a ‘preview’ component in VisualWorks 8.2

Significant refactoring of SiouX server

Prerequisites:

- Application Layer Protocol Negotiation (**ALPN**)
- New TLS Cipher Suites (**AEAD**)

HTTP/2 cont'd

ALPN

- Initial request comes over HTTP/1.1
- ALPN allows protocol upgrade to HTTP/2

HTTP/2 will work over plaintext connection, BUT:

All web browser vendors have stated their intention to support HTTP/2 only over TLSv1.2 or higher

Requires TLS_ECDHE_RSA_WITH_**AES_128_GCM**_SHA256 cipher suite

The **AES_128_GCM** part is a challenge…

Security

AES_128_GCM encryption is a different type than used in TLSv1.1 or earlier

Authenticated **E**ncryption with **A**dditional **D**ata (**AEAD**)

Encrypt/decrypt operation occur in-place

Plaintext (input) buffer and cipher text (output) buffer MUST be identical

Significant difference compared to other encryption algorithms

Currently works with OpenSSL 1.0.x, in VW 8.2

Security cont'd

Older security protocols and algorithms are constantly under scrutiny

The **RC4** cipher has been found insecure in both theory and practice, i.e.: it is **broken**.

Leading vendors (Microsoft, Apple, etc…) have removed RC4 support from their products

RC4 no longer available in VisualWorks 8.2

If your software requires RC4, you should make changes before upgrading to VW8.2

LDAP(S)

Lightweight **D**irectory **A**ccess **P**rotocol

LDAPv3 specification allows using a **TLS** extension to the protocol, using a **StartTLS** request to the server -> **LDAPS**.

LDAP in Open Repository since 2001.

Some LDAP refactoring was necessary in order to make LDAPS work.

Cincom will maintain LDAP in the Open Repository

LDAPS is only available as part of VisualWorks distribution.

Future work

First priority: Continue to respond to our **Customers' needs**. (past examples: MASSL, LDAPS)

Keep improving tools & integration (e.g., JavaScript auto-complete)

Expand external library security support (e.g., bcrypt.dll, OpenSSL 1.1.x when final)

Make **Smalltalk** the best environment to develop complex, interconnected software **ecosystems**

Questions?

Thank You!

Suzanne Fortman
Program Director / Engineering Manager
sfortman@cincom.com
@SuzCST (Twitter)

Arden Thomas
Product Manager
athomas@cincom.com
@ArdenTCST (Twitter)

Jerry Kott
jkott@cincom.com
@cincomsmalltalk (Twitter)