Dojo 1.0 : Doubling Down On The Open Web

Alex Russell
Project Lead, The Dojo Toolkit

presented by
sitepen

A Bit Of Background

In The Beginning Were The Hacks...

- Early libraries didn't have a natural audience
  - HTML + CSS not fully explored
  - Browsers still in flux
  - Wide capability gaps required too much code to paper over
- Most JS authored in strict proceedural style
- Purely decorative DHTML the norm

A History Of The Future

- "Netscape will be the only browser!
  It will displace windows!" *
- "The DOM will unify all the browsers!
  Just write to the W3C DOM"
- "XForms will make this all irrelevant"
- "XHTML is semantic enough...you just need CSS"
- "Ajax will give us what the browsers won't"*

* followed by "the Internet changes *everything!!*"

Dojo 0.4.x Planned For Many Futures

- Abstract widget classes allowed rendering in multiple environments *
  - HtmlWidget
  - SvgWidget
  - SwtWidget
- Abstract parser provided capacity to parse XML or HTML *
- Package system and bootstrap pluggable

* in theory, but no significant number of users ever used these features

Futurism Is A Bad Predictor Of The Future

Past Performance *Is* An Indicator of Future Returns

Things We Can Expect

- IE still upper limit of web's expressive capacity for 2+ years
- No vendor likely to win new Browser Wars enough to justify not building for others
  - IE 6 with us for some class of apps indefinitely
- Plain-text formats will carry the day
- JavaScript extension points likely to evolve faster than widespread use of tags
- Interop with mobile assumed

More Things We Can Expect

- Users will continue to want richer experiences
- Markup-driven UI construction will still win
- Processors will get faster
- Bandwidth will improve
- More ram will be installed
  - only a small percent available to us
- Latency will still suck
  - Fixed storage on meteoric size trajectory
  - Fixed storage latency essentially fixed

Things We Don’t Know Yet

- When will we be able to drop IE 6 (and 7)?
- Will ES4 make inroads? When?
- What will browser support for HTML 5 tags be? When?
- What HTML 5/WhatWG JS APIs will be introduced by IE.Next?
  - storage?
  - offline?

Planning For All These Futures Is Futile

sitepen

The Common Thread: When?

What To Do?

Designing With Play In The Joints

- Orthogonal, modular design
- Must work well in current constraints to win market share
- Must allow others to easily plug in high-order functionality when conditions allow
- “Open” not a panacea
  - But a huge advantage when in striking distance of feature parity

HTML *Is* The App Container

“Simplicity does not precede complexity, but follows it.”

Alan Perlis

1.0 Design Goals

- Remove the magic
- Pave the fast paths
- Fewer idioms, consistently applied
- Every K should benefit the user
- Upgrade the web we've got, don't replace it

- *Provide more headroom*

Immediate Concerns: Widgets

- Yours!
  - Dojo widgets not more “blessed” than yours
  - Pre-built pieces for mixing in
- Ours
  - a11y
  - i18n
  - Themes
  - Performance

Immediate Concerns: Packaging

- Always available, now with better flexibility
- Develop in the large with less worry
- Deployment optimization for your code, not just ours
  - Package system handles dependencies
  - “Layered” builds
  - Re-written build system
    - Ant is dead! Long live Rhino!

Immediate Concerns: Data Access

- dojo.data provides unified API
- Data consuming widgets dojo.data savy
- Quickly expanding list of providers
  - JSON
  - XML
  - CSV
  - Paged JSON via REST queries
  - HTML tables
  - OPML
  - Flickr, Picasa, other service endpoints

Immediate Concerns: DOM Scripting

- Building blocks now easily separable
- “dojo.js” always means the same thing
- dojo.query() now always available
- dojo.behavior, based on dojo.query()
- dojo.connect() normalizes even better

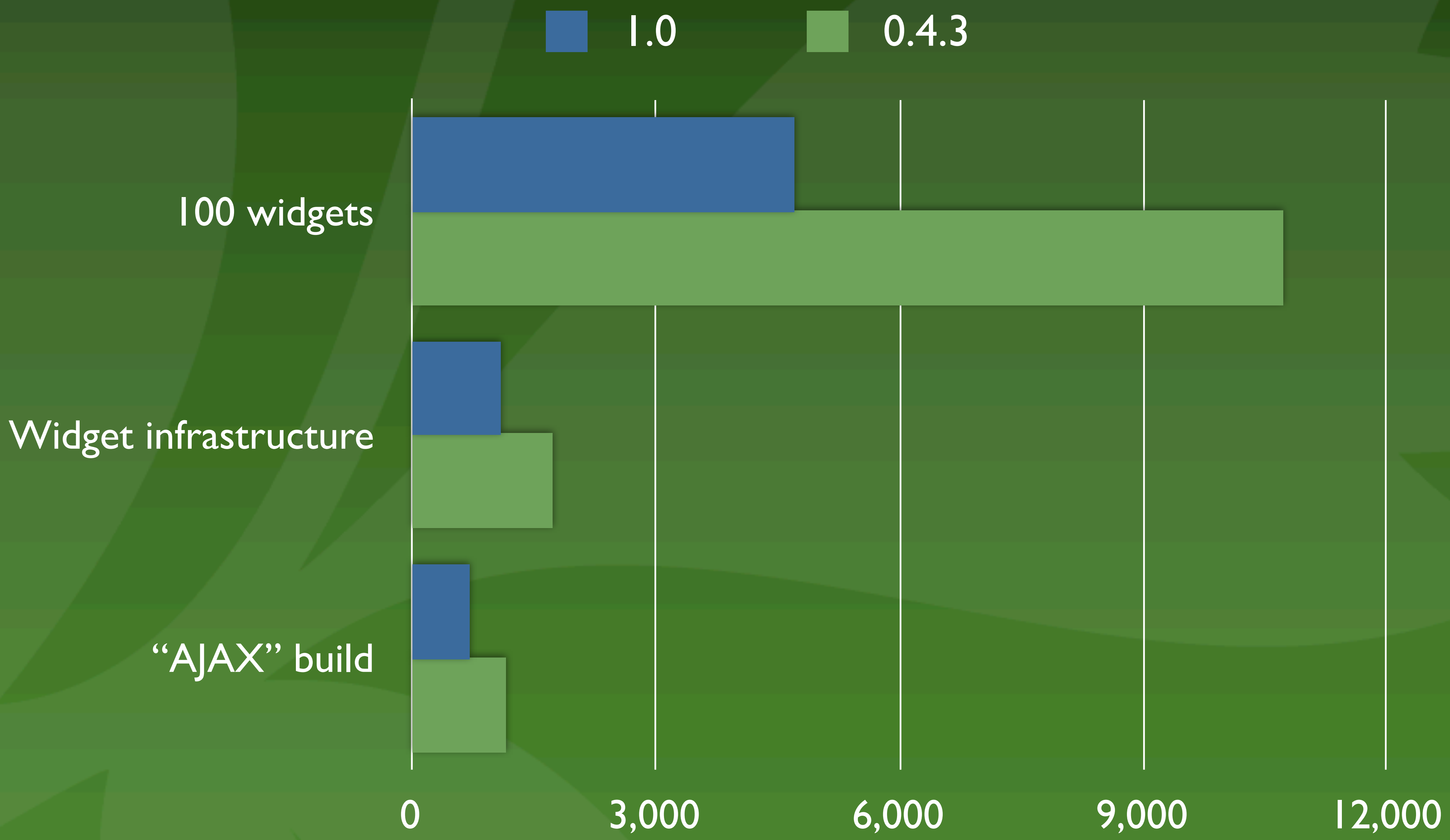
Immediate Concerns: Performance

- Core is 40% smaller than 0.4 Ajax (25K vs. 40K)
- Does more in less space
- Core+Widget infrastructure is 35% smaller
- Widget creation is significantly faster (2-5x)
- Similar tasks allocate half as many objects

Build Sizes

Variant	0.4 Ajax	1.0 Base
Uncompressed	319K	218K
Built	148K	50K
Gzipped	39K	23K

Object Allocation



Speed/Size Improvements Enabled By Removing Unused Flexibility

On The Cusp

- Grid Widget
  - Sortable, data-bound, editable
  - Virtual scrolling
  - Locked headers and columns
  - Themes
- Charting
  - Based on dojox.gfx 2D and 3D layer
  - Supports many common chart types
  - dojo.data backed

On-The-Cusp Features Mine Ubiquitous But Divergent Substrates

Future Concerns: Feature Ubiquity

- Cost-to-implement is a market-making concern
- Uneven feature terrain means many features latent and unexplored
  - See: Ajax
- Next up:
  - Mobile profile
  - Storage
  - 3D drawing, charting
  - Offline
  - Audio/Video

Dojo 1.0: Better Experiences, Smarter Upgrades 100% Open Web

dōjō

sitepen