



Lighting the Shadows

A Roundtable on Overcoming Fusebox Objections



Background

- Many CFers aren't making the move to FB
 - Co-workers, superiors, respected associates may not yet be persuaded, may be antagonistic
 - Watercooler or email discussions taking place
 - Are all participants fully informed?



Background

- Heated discussions take place even within FB community
 - May be on public mailing lists, or only among FB architects
 - Even if publicly available, will prospective FBERs really take the time to read these?
 - Are there known issues that aren't often publicized?
 - Perhaps not seen as significant by some

Bottom Line

- Are there valid points of contention about FB?
 - Problems, challenges, weaknesses?
 - Work-arounds? Opportunities for improvement?
- Valuable benefits that people just don't get?
 - Ways that FB makes life easier, at a modest cost of learning a new approach, that aren't widely known
 - Is there a PR problem?



Premise of RoundTable

- Frank, no-holds-barred discussion
- Share observations about the discussions taking place in the CF community
- Inform each other of the strengths/weaknesses of these arguments
- Perhaps form basis of new whitepaper



Possible Outcome: Whitepaper

- Perhaps “Selling Fusebox to Skeptics”, outlining:
 - Broad pros and cons of FB
 - Frank admission of any known challenges
 - List any work-arounds available today or in FB3
 - Discussion of any long-term FB improvements to address those issues



Panel Participants

- Moderator: Charlie Arehart
- Panelists:
 - Steve Nelson
 - Nat Papovich
 - Gabe Roffman
- Audience participation especially welcomed

Pros

- Some not unique to FB, but promoted by it
 - Specification, design up-front
 - Wireframes, FuseDocs, FuseML
 - Self-documentation, maintenance by others
 - Encapsulation, component-orientation, reuse
 - Abstraction, inheritance
- Concepts help some transitioning to Java/JSP
 - Indeed, FB is being proposed to work for that as well

More Pros

- FB-specific features
 - Segmentation of display, query, action files
 - From each other, so changes can be isolated
 - With naming that is self-documenting
 - Development can take place independently
 - Reusable within and among applications
 - As are fuses, circuits and applications

More Pros

- FB-specific features
 - Inheritance (of layout, variables, etc.)
 - Single-point of entry
 - Works with small or large apps
 - Dissuades changes after design

Some Cons

- Documentation
- Framework, methodology, architecture, or best practice?
- Single-point of entry challenges
- Search engine issues
- Supporting calling as custom tag, when not used
- Heavy use of CFINCLUDEs

Some Opportunities for Improvement

- More built-in features?
 - Error handling
 - Security
 - Designation of flow of control among fuses
 - Detecting cookies, passing urltoken if needed