Application Communities

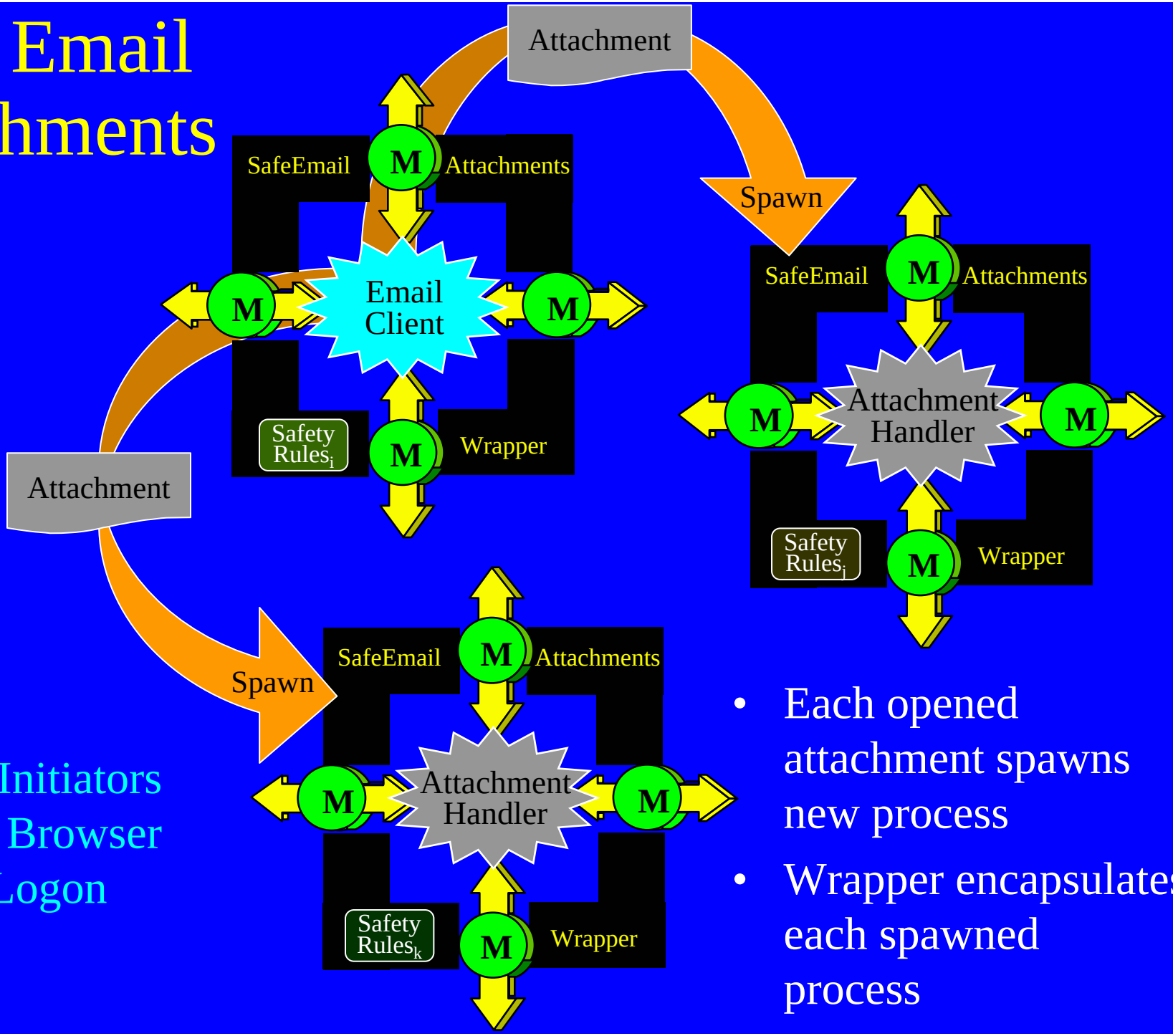
Bob Balzer

Teknowledge

balzer@teknowledge.com

TEKNOWLEDGE

Safe Email Attachments

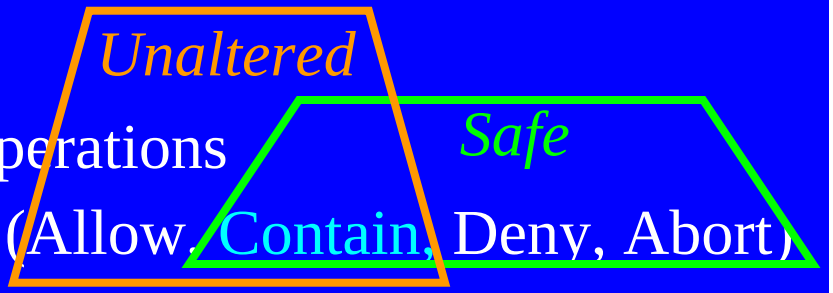


Other Initiators
- Web Browser
- WinLogon

- Each opened attachment spawns new process
- Wrapper encapsulates each spawned process

Contained ~~Wrapped~~ Execution

- Goal: Safely Execute possibly Malicious Code
- Approach:
  - Mediate potentially harmful operations
  - Apply Authorization function (Allow, Contain, Deny, Abort)
  - Contained operations only affect wrapped process
- Problems
  - Configuration ~~difficult~~ easy { *Allow* desired changes; *Contain* everything else }
    - Tight policy generates many false positives
    - Loose policy leaves room for undetected malicious activity
  - ~~Early~~ Late authorization decision ~~required~~ allowed (after execution)

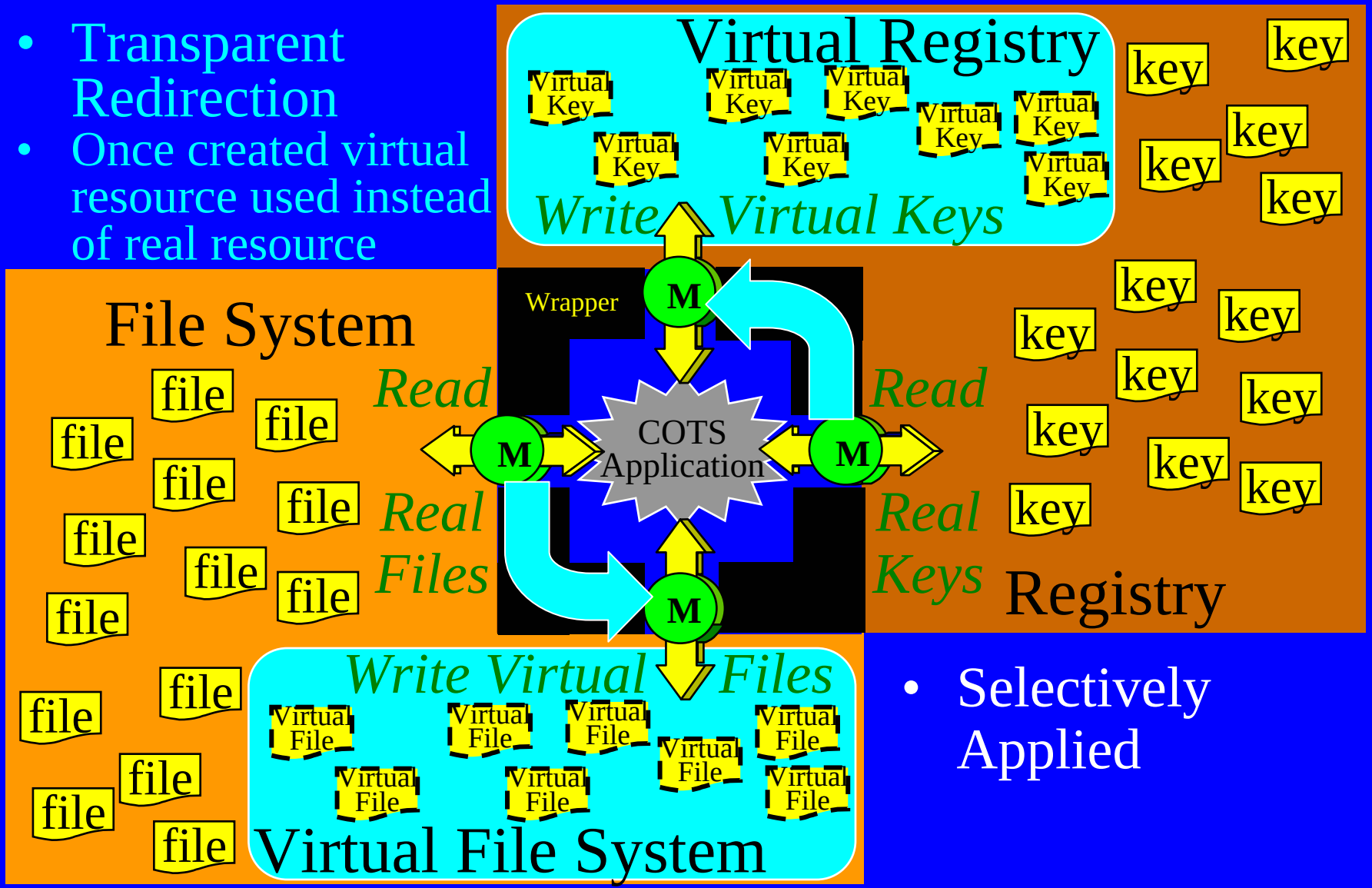


Desired Changes

Attachments => None
Editors => Edited document

Contained Execution

- Transparent Redirection
- Once created virtual resource used instead of real resource



- Selectively Applied

Contained Execution

- Like a Virtual Machine
  - Execution is isolated
- Unlike a Virtual Machine
  - Process-Level (instead of machine-level)
  - Selective (instead of copying entire environment)
  - Incremental & dynamic (copies created as needed)

Contained Execution
(contain selected modifications within process)

- Contained Resource (currently implemented)
  - Virtual Registry (selected changes made to virtual keys)
  - Virtual File System (selected changes made to virtual files)
- Benefits:
  - Program Execution has no effect on rest of system
    - ⇒Blocks single-stage attacks (no effect on rest of system)
    - ⇒Blocks multi -stage attacks (no transfer of aggregated effects)
  - Rule violations can be safely contained and auto-authorized
  - Attack determination can be safely delayed
    - More behavior analyzed => better decision
    - Supports autonomic responses
    - Reduced false alerts
  - Can rerun information extraction attacks with misinformation

Contained Execution Demo

Block Infection Propagation via Contained Execution

- OUT From Corrupted Application Instance
  - to other Application Instances
  - to other “carriers”
  - Corruption path is contained via virtual resource
    - Inside change doesn’t affect outside resource
- IN to uncorrupted Application Instance
  - from other Application Instances
  - from other “carriers”
  - Corruption path is contained via virtual resource
    - Outside change doesn’t affect contained virtual resource