Problems running untrusted services as Java Threads

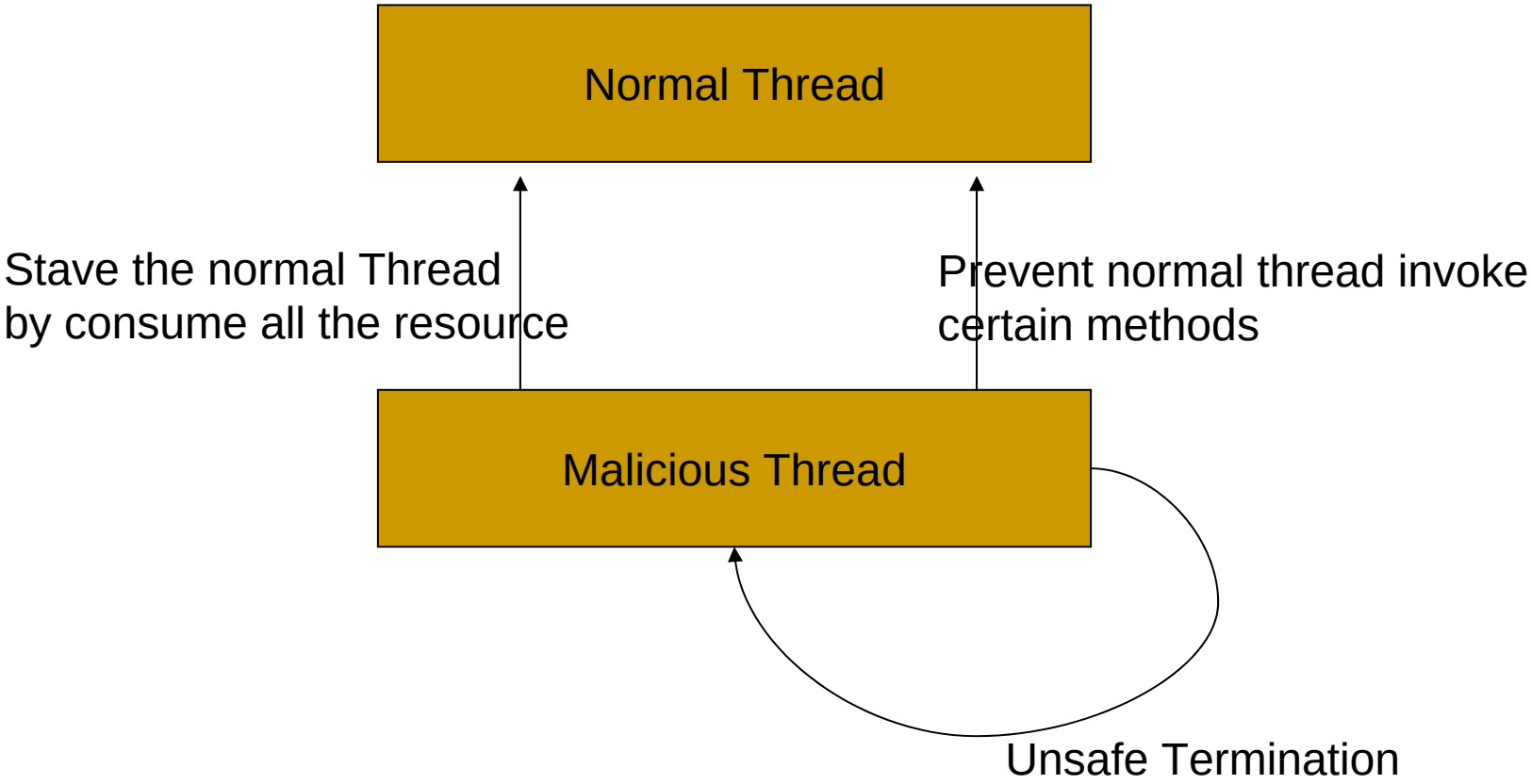
[Ap He05] A. Herzog, N Shahmehri, "Problems Running Untrusted Services as Java Threads", in *Certification and Security in Inter-Organizational E-Services*, IFIP 18th World Computer Congress, ed. Nardelli et al., Aug 2004, pp. 19-32.

Presenter: Jason Chen

Summary of this paper

- Problem: A number of JAVA environments run untrusted services as JAVA threads which can cause other threads dysfunctional
- Solution: This paper try to solve the problem in 3 Directions
  - Safe Termination
  - Resource Control
  - Isolation

JAVA Thread Security Model



Suggestions to solve security problem

- More resource control
- More Access control
- More defined security policy

Appreciative comment

- Author gave a very detailed explanation on how to solve the security problem between normal thread and malicious thread
  - By restricting the interaction between threads to limit the damage which can be caused by malicious thread
  - This is useful to other thread models
  - This could relate other real life security problem
    - Eg. Treat different departments in the company as threads

Critical Comments 1

- A statement that could lead to confusion

  "..if reliably stopping threads is an issue for the container, one should consider running the untrusted code as separate process in its own JAVA virtual machine.." page 26

  - Author never mentioned the reason why we only have one JVM
  - That is what normal JAVA thread system is doing
  - Efficiency trade off

Critical Comments 2

- The whole paper is talking about damage control rather than how to avoid the damage.
  - Why do we have to run untrusted service as JAVA thread at first place?
  - Already suffered loss once the malicious thread get in successfully

Question

- Given the fact:
  - Author suggests restrict the interactions could improve security
  - Restrict the interaction could reduce the performance of Java threads
- Do you think we should use author's suggestions to improve the security by sacrificing the performance?