Content Distribution, Rights Management and Trusted Platforms

Brian A. LaMacchia
Software Architect
bal@microsoft.com

Windows Trusted Platform Technologies
Microsoft Corporation

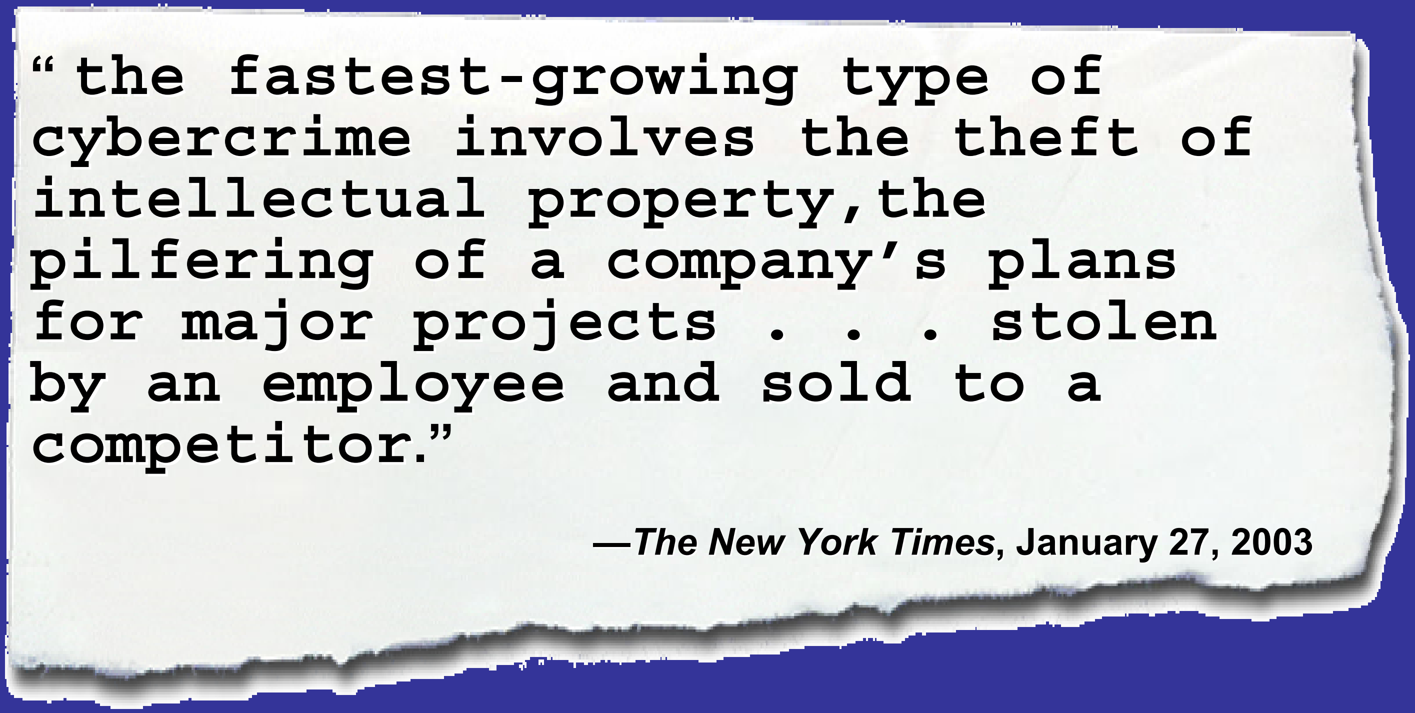
CPSC 155b
E-Commerce: Doing Business on the Internet

March 25, 2003

What is Content Distribution?

- The movement of content (any digital information) across a network from the content creator's machine to a content user's machine
  - Usual example: electronic distribution of mass-market media (books, music, movies) from the content creator (or licensee) to the consumer.
  - But enterprises have similar situations
  - Consumers also distribute content
    - Personally-identifiable information

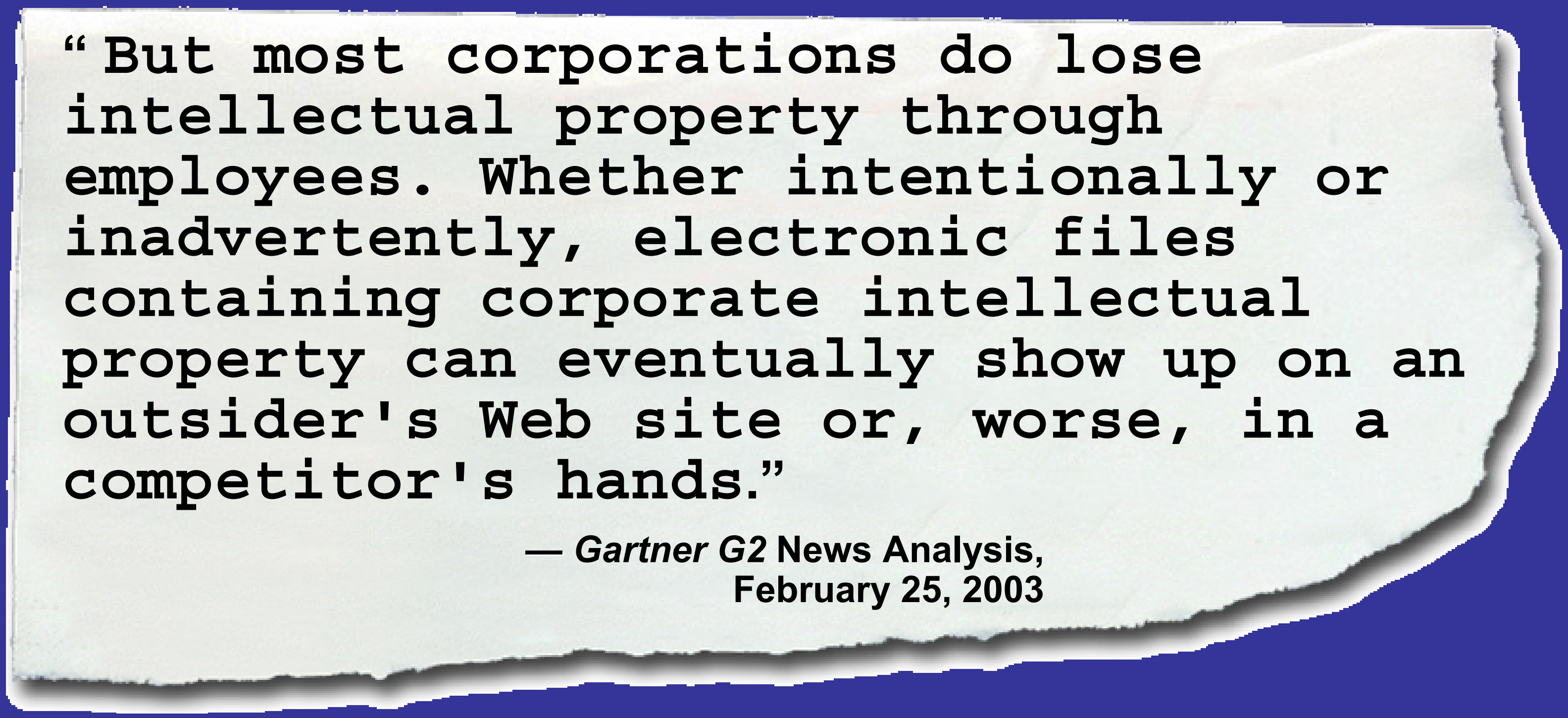
Enterprise challenges



> “ the fastest-growing type of cybercrime involves the theft of intellectual property,the pilfering of a company’s plans for major projects . . . stolen by an employee and sold to a competitor.”

—*The New York Times*, January 27, 2003

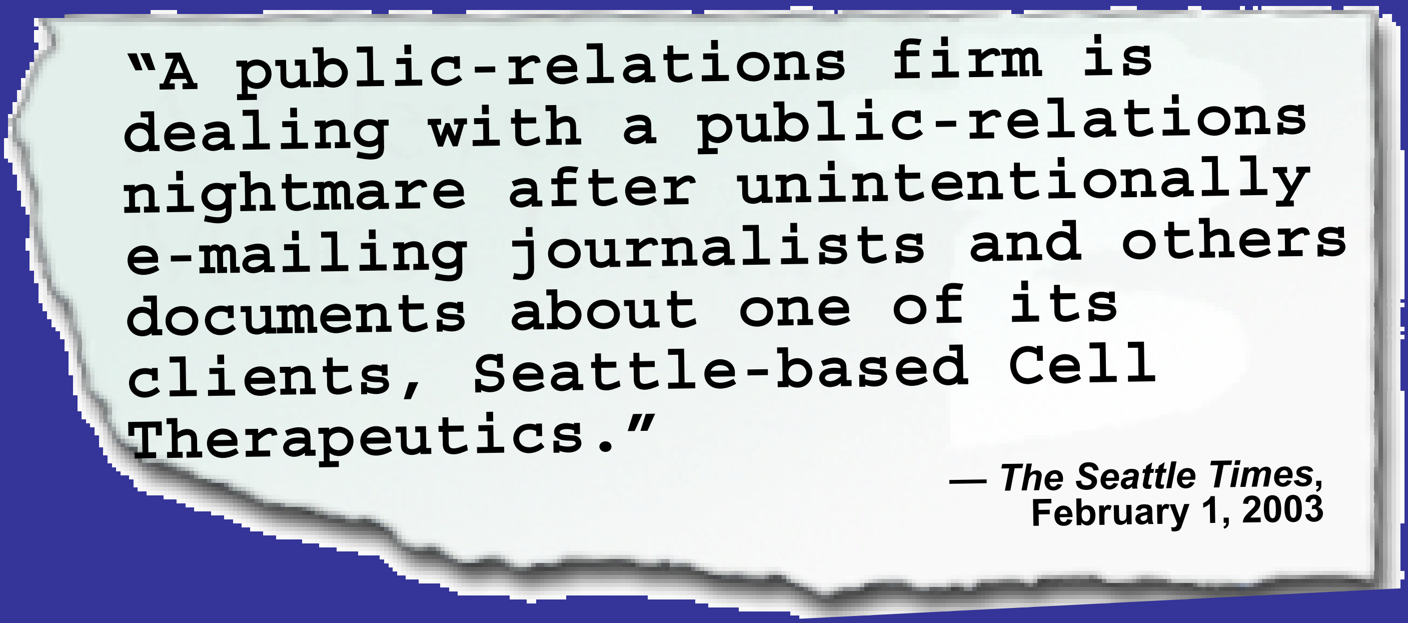
Enterprise challenges



“But most corporations do lose intellectual property through employees. Whether intentionally or inadvertently, electronic files containing corporate intellectual property can eventually show up on an outsider's Web site or, worse, in a competitor's hands.”

— *Gartner G2* News Analysis,
February 25, 2003

Enterprise challenges



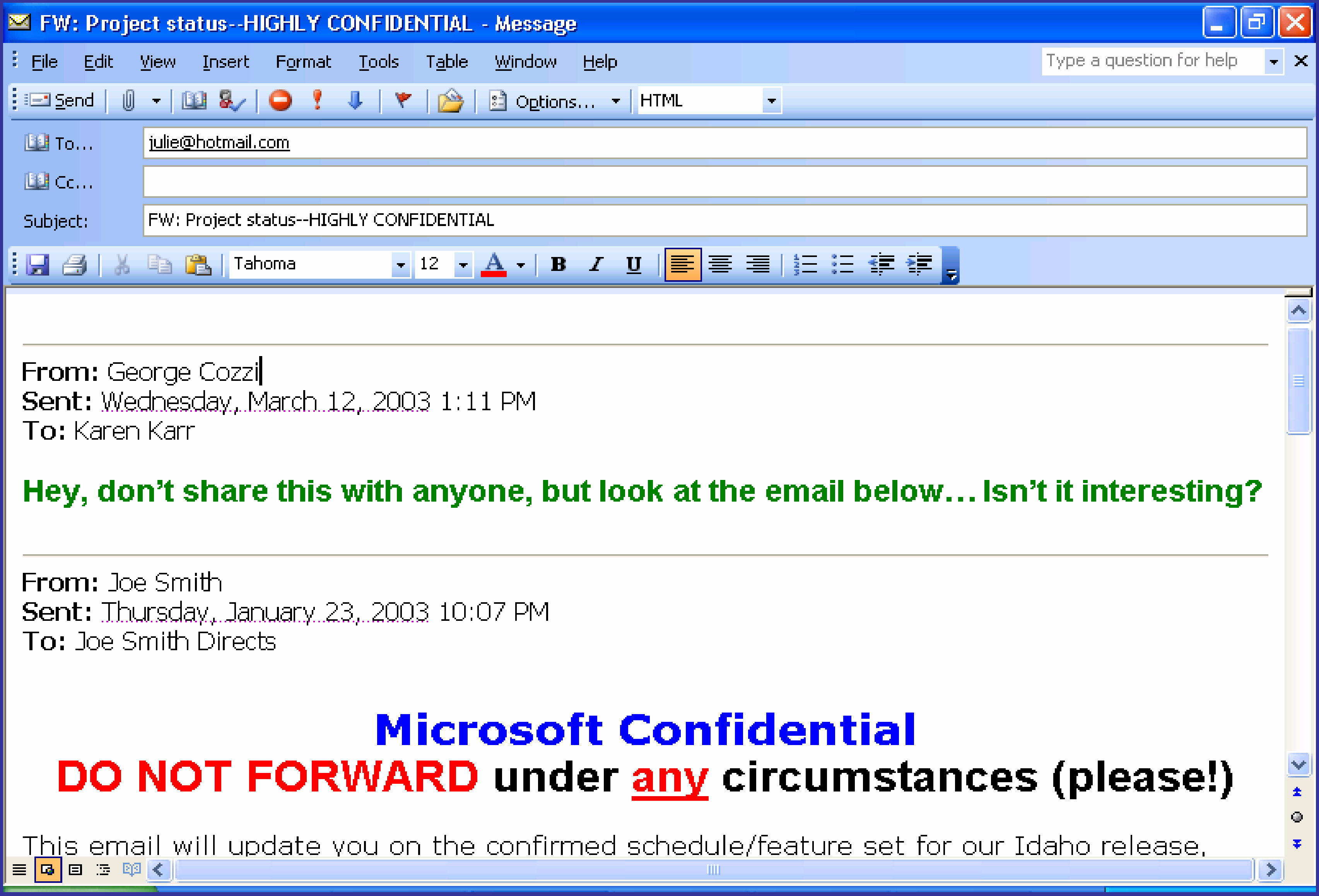
"A public-relations firm is dealing with a public-relations nightmare after unintentionally e-mailing journalists and others documents about one of its clients, Seattle-based Cell Therapeutics."

— *The Seattle Times*,
February 1, 2003

Enterprise challenges

- 32% of the worst security incidents were caused by insiders; 48% in large companies
- Intellectual property theft caused the greatest financial damage of all security failures
- Costs:
  - Consultant fees (fix damage), Down time, Brand damage, Legal liability, Customer confidence, etc.

Have you ever encountered this?



To... julie@hotmail.com

Cc...

Subject: FW: Project status--HIGHLY CONFIDENTIAL

From: George Cozzi
Sent: Wednesday, March 12, 2003 1:11 PM
To: Karen Karr

Hey, don't share this with anyone, but look at the email below… Isn't it interesting?

From: Joe Smith
Sent: Thursday, January 23, 2003 10:07 PM
To: Joe Smith Directs

Microsoft Confidential
DO NOT FORWARD under any circumstances (please!)

This email will update you on the confirmed schedule/feature set for our Idaho release,

What is the Content Distribution Problem?

- The content distribution problem is one of control.
  - Content creators want (some) control over how their content is consumed after it leaves their machines.
- The Internet does a great job moving bits around, but it cannot enforce policies on those bits once they've moved.

Rights Management

- Enter rights management (RM) systems, which aim to associate policy with content as that content flows across networks and enforce that policy at remote nodes.
  - The term “digital rights management” (DRM) tends to be used today to refer to RM systems specifically designed for mass-market media.
  - We’re going to talk about the general RM case today, not specifically RM for media

Agenda

- Motivation: RM as a policy management problem
- Rights expression languages – XrML 2.X
  - Authoring & evaluating policies for use of content.
  - For DRM: representing grants of copyright-related rights and modeling physical-world transfers of rights.
- Trusted Computing Platform Alliance (TCPA) and Next-generation Secure Computing Base (NGSCB)
  - Approaches to adding "attestation" capabilities to the PC.
    - Attestation allows the PC to make a digitally-signed statement about its state (e.g. that some set of software is currently running).
  - Projecting policy expressions with confidence into remote environments.
  - For DRM: content owners can have some assurance that recipients will abide by their policies for use of their content.

Motivation

- When we think about RM systems, we tend to focus on the content to be managed.
  - How is the content protected/encrypted?
  - How are the keys managed?
- In this lecture I want to focus not on the content but on the policies we associate with content.
  - Think about RM systems as they relate to policy expressions.
  - How are content policies written, distributed and enforced?

Policy Enforcement Systems are Prolific

- When you view RM as a policy distribution & enforcement mechanism, you find lots more of them exist than you might expect…

Policy Enforcement Mechanisms in Microsoft Products Today

- MS DRM for eBooks
- MS DRM for Windows Media
- Windows Rights Management Services
  - Office 2003 Information Rights Management
- License servers for Terminal Services, File & Print Services, etc.
- Xbox (anti-repurposing)
- Ultimate TV/eHome (digital storage of video)
- File system ACLs
- Enterprise policy management
  - Group policy in domains
- Partially-trusted code policies (.NET Framework)
- NGSCB

Policy-related Tasks in RM Networks

- Content owners (or their agents) author policy statements for content.
  - Owners license their exclusive rights (in a copyright sense) to consumers or distributors.
- RM-aware servers (or networks) distribute policy statements.
  - Maybe they distribute the content too.
- End-user RM systems consume and abide by policy statements when processing the content.

Key Technical Challenges

- As an industry, we understand the “crypto” aspects of RM better than we understand the “policy” aspects.
  - Key management is easier than policy management.
- Critical “policy” work areas include:
  - Authoring & evaluating policy expressions
  - Projecting policy expressions with confidence into remote environments

General description of RELs

- A rights expression language (REL) is a type of policy authorization language.
  - Focus is on expressing rights granted by one party to another.
  - Issuance and delegation rights for other grants are core concepts.
    - Can be used to model lending, loans, transfers of rights.
- REL design goals:
  - Provide a flexible, extensible mechanism for expressing authorizations.
  - Enable interoperability across various policy evaluation systems.
  - Make it easy for policy authors (e.g. content owners) to express their desired policies.

An example REL: XrML 2.X

- XrML, the *XML Rights Management Language*, is a standard currently under development

XrML 2.X

- In the RM context, XrML 2.X allows content owners a systematic way to express their intent for distribution and consumption.
- Like other policy languages, XrML 2.X licenses (statements) declare authorizations, but cannot enforce compliance.
  - Systems that consume XrML 2.X licenses must be trusted by the license issuer to properly enforce the grants specified within the license.
- Licenses are digitally signed by the issuer to protect their integrity.
- Licenses may be embedded within content or move independently.

Semantic of a Grant

- Every XrML 2.X grant has the following form:
  - Issuer authorizes principal to exercise a right with respect to a resource subject to conditions.
  - A license is a collection of one or more grants made by the same issuer.
- Grants may be chained together:
  - Bill’s RM system trusts Tom and his delegates.
  - Tom delegates the right to license printing to John.
  - John issues a license: “Bill has the right to print the book.”
  - Therefore Bill can print the book.

Sample XrML 2.X License

```
<?xml version="1.0" encoding="UTF-8" ?>
  <license>
  <grant>
    <keyHolder>  …  </keyHolder>
    <mx:play />
    <mx:diReference>
    <mx:identifier>urn:mpeg:example:2002:twotonshoe:album</mx:identifier>
    </mx:diReference>
  </grant>
  <issuer> … </issuer>
</license>
```

XrML authorization model

- Input
  - Principal
  - Right
  - Resource
  - Time interval
  - Licenses
  - Designated “root grants” (implicitly trusted)
- Output
  - “No”
  - “Yes,” unconditionally
  - “Maybe,” if a set of conditions are also met

XrML Key Language Features

- Mechanisms for enhanced expressivity
  - Patterns, variables and quantifiers
  - Grouping grants
  - Delegation
- Meta-rights
  - Issue
  - Obtain
  - Revocation
  - PossessProperty
- Linking conditions
  - PrerequisiteRight

XrML 2.X and Multiple Authorities

- XrML 2.X offers a new level of expressiveness
  - Enables representation of a wider range of scenarios.
- Example scenario: evaluating authorizations from multiple authorities for a resource.
  - Today, RM systems operate using a “closed-world assumption.”
    - Any action not explicitly authorized by the content owner is prohibited.
  - Copyright doesn’t work like this.
    - Copyright is a liability-based system.
    - Some actions are permitted by law even if they are not explicitly authorized by the copyright holder.
  - How might we use XrML 2.X to represent authorizations as well as limitations built into the law?

XrML 2.X and Multiple Authorities (cont'd)

- Content creators are given exclusive rights by law; these rights are then licensed to consumers.
- Limitations on the exclusive rights contained in a copyright can be thought of as independent grants of licenses by Congress to the consumer.
  - "Congress says every library has the right to make an archival copy of a work" (17 U.S.C. 108).
  - Variables allow us to write licenses that apply to (potentially undefined) sets of content and users.
  - Congressional grants can be conditioned on possession of a licensed copy of the work.
- RM systems would need to recognize both the content owner as well as Congress as authorities for a given work.

Evaluating Policy Expressions

- RM systems attach policy expressions to content and then project that policy along with the content into a remote system.
  - Policy creators need to have confidence that the receiving system will faithfully implement the defined policies.
- For years in security research, we've built protocols that depend on trusted computing bases (TCBs) at their core.
  - The TCB must behave as expected, because it's the part of the system which you have to implicitly trust.

Attestable TCBs

- For RM systems, having a TCB locally is not sufficient to ensure very high levels of trust
  - We need to be able to prove the existence & reliance on a TCB to a remote party.
  - “Attestation”
- A content author is only going to allow content & policy to flow to TCBs (and, recursively, applications) he believes are going to behave properly.
  - “Behave” == implement policy as defined
- Content consumers are only going to let code they understand run their systems.

Trust is Central to Attestable TCBs

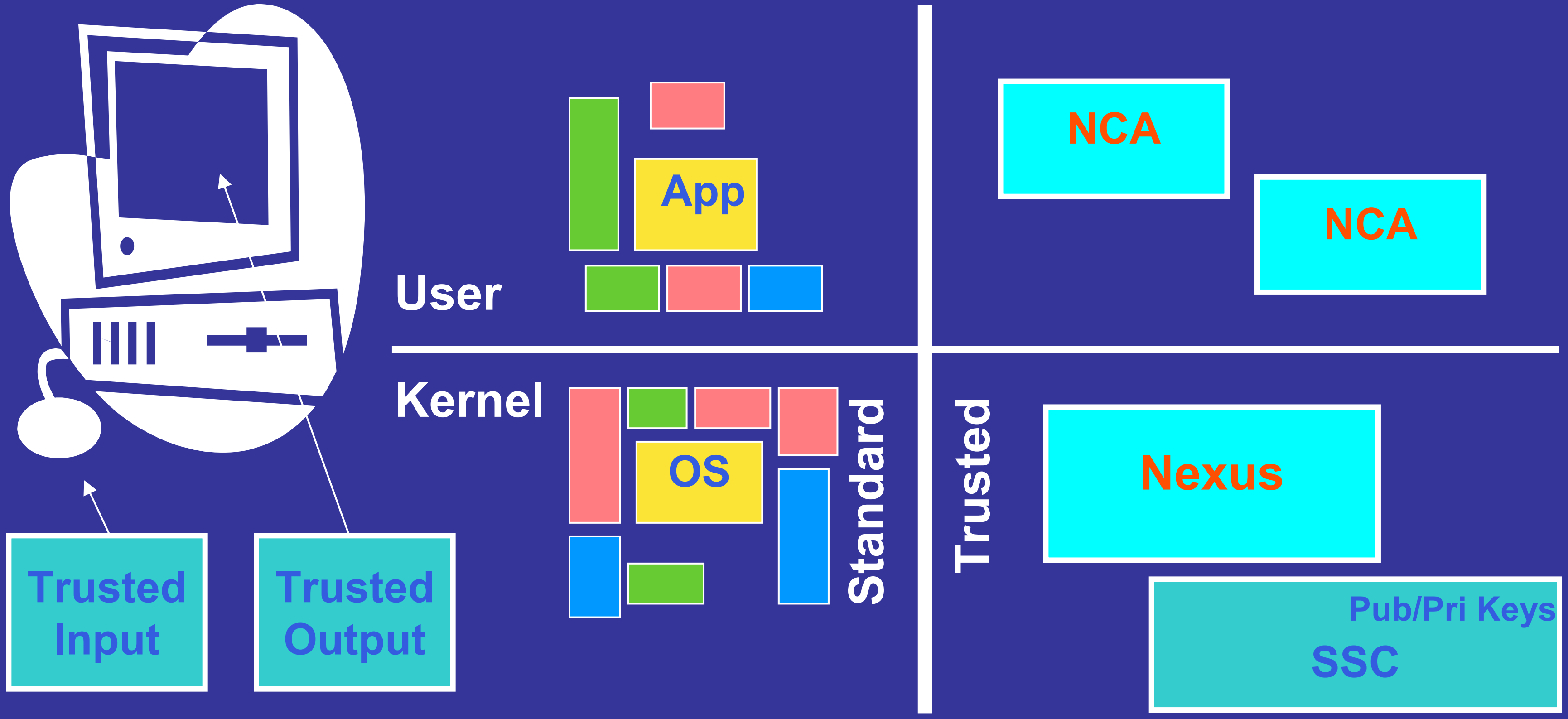
- Four elements that must be present in order to trust a TCB
  - I know who / what the it is, and that it is not an imposter
  - I know its state – it has been properly initialized
  - I know that it cannot be tampered with
  - I know that my communication with it is private and tamper-proof

Building Attestable TCBs

- There are two separate industry initiatives today trying to build attestable TCBs on evolutions of PC hardware and software.
  - TCPA – Trusted Computing Platform Alliance
  - Microsoft’s Next-generation Secure Computing Base (NGSCB)
- TCPA is specifying changes to the PC hardware that can make attestations.
  - Goal is to be able to sign statements about the entire software stack running on top of the PC, from the moment power is turned on forward
- NGSCB has a somewhat different focus.
  - Goal is to create a separate, parallel execution environment inside PCs that is rigidly controlled by the user, and make attestations about only that code.
  - Additionally, provide sealed storage, curtained memory and secure I/O with the user.

NGSCB – How It Works

- Subdivide the execution environment by adding a new mode flag to the CPU.



Attestation in NGSCB

- Attestation is a recursive process
  - The SSC (security chip) always knows the unspoofable identity of the running nexus.
  - Assuming it does, the SSC can then attest to (make signed statements about) the identity of the nexus.
    - SSC has a digital signature key pair, plus some certificates for that key pair.
  - The nexus in turn can attest to the identity of nexus computing applications (NCAs)
- If you accept the certificates & digital signature key pair as belonging to an uncorrupted SSC, then you can trust the statements the SSC makes about the running nexus.

Attestation and RM Systems

- Why would RM system builders be interested in the attestation feature?
  - Attestation allows a host machine to query what software is running on a remote machine before sending it content.
- Examples:
  - In an enterprise RM environment, servers could be configured to only release classified documents to non-portable machines.
  - Before sending personal information to a server, a client could demand proof that the server is running a software stack certified to comply with privacy-protecting principles.
  - In a consumer RM environment, content could be licensed such that it could freely migrate among all devices within a single “household”.
- Operation of the PC is never blocked; the hardware simply will not lie about the software running on top of it.
  - Servers can choose not to talk to clients they don’t like.

Summary

- Two security technologies:
  - Rights expression languages (RELs)
  - Attestable TCBs
- These technologies provide a number of new security features for computing platforms, including advances in secret storage and policy expression, evaluation and projection.
- RM systems built on today's platforms are useful for a wide variety of solutions; the features provided by RELs and attestable TCBs will further expand that set.

Questions?