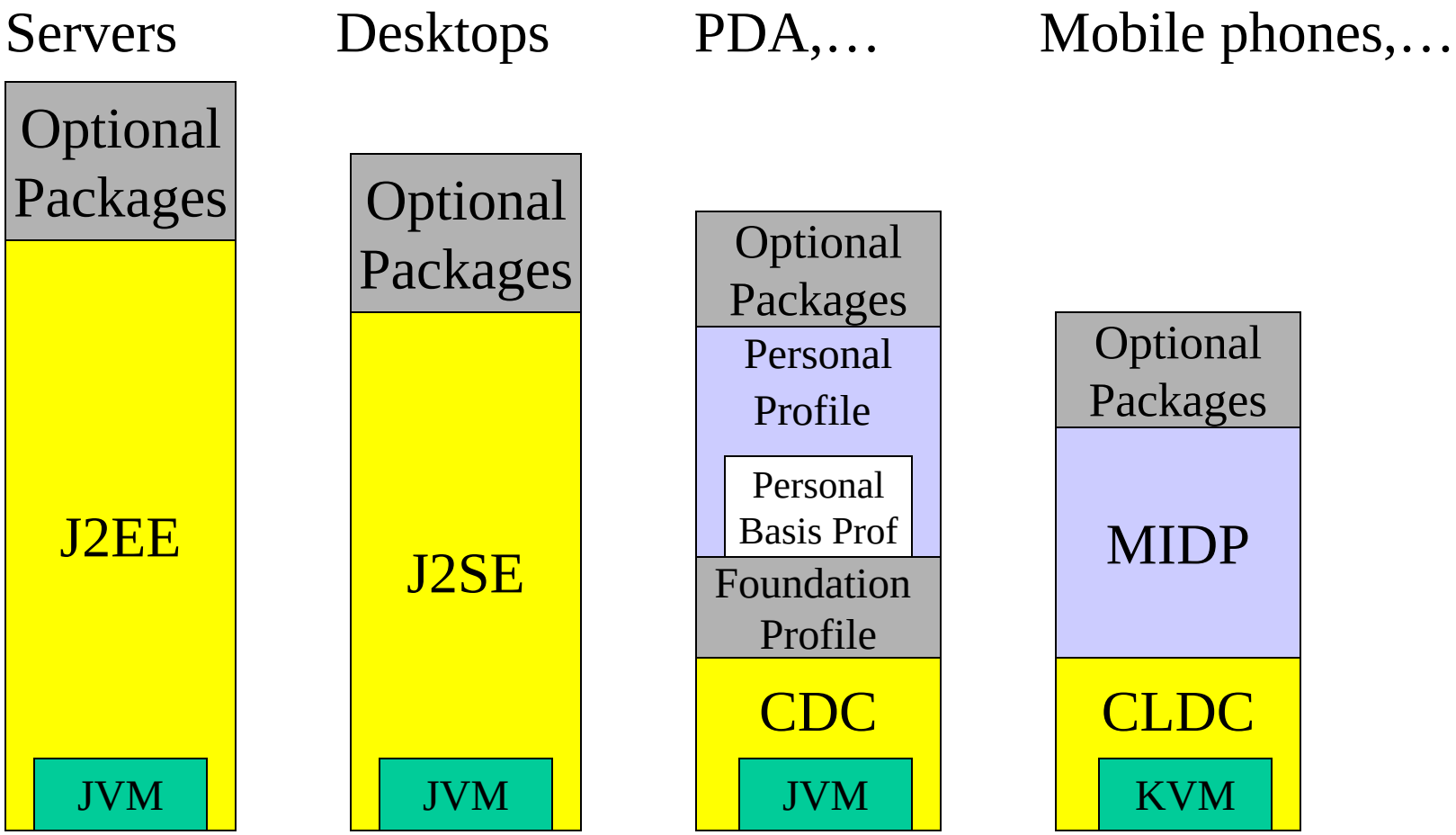
Lecture 16
CS 2506

MIDP Security

Java 2 Platform



Configurations

- It is necessary to provide a set of base classes appropriate to each group of mobile devices.
- One configuration consists of a VM and a minimal set of class libraries chosen to provide the base functionality for a distinct set of devices.
- Connected Device Configuration (CDC) is designed for resources-rich devices.
- Connected Limited Device Configuration (CLDC) is intended for devices with less resources.

Profiles

- Whereas a configuration provides the lowest common denominator for a group of devices, the profile brings APIs for a specific class of devices. For example, as some mobile phones have more resources than others, they can offer more in terms of the interface to the hardware.
- Currently, there are four profiles, only one of them being a CLDC profile.
  - Mobile Information Device Profile (MIDP): UI, network connectivity, local data storage, application lifecycle mgmt;
  - Foundation Profile (with CDC): network-capable, no UI;
  - Personal Profile: full Abstract Window Toolkit (GUI), web;
  - Personal Basis Profile: is a subset of the Personal P – limited GUI

The Mobile Information Device Profile - MIDP

- Standard classes for user interface – screen objects, forms, all device-aware;
- Multimedia and gaming – Mobile Media API (MMAPI), Game API (provides a manager for sprites and layers);
- Extensive connectivity – http, https, datagrams, sockets, serial ports, SMS (GSM, CDMA), event-based networking model;
- Over-the-air provisioning (OTA) – deployment and updating of applications over the air (apps are discovered, installed and removed on MIDP devices);
- Persistent storage – record-based database mgmt system;
- End-to-end security – https for transmission of encrypted data; security domains are used to identify trusted and un-trusted MIDlets.

The MIDP Model

- A MIDP application is called MIDlet.
- The MIDP model defines how the MIDlet is packaged, what runtime environment is available and how it should behave with respect to the constrained resources of the device.
- MIDlets can be packaged together in suites and share one another's resources – graphics, data records.
- Each MIDlet suite also has a descriptor file, JAD, which allows the Application Mgmt Sw (AMS) to identify what is about to install prior to installation.
- The model also defines a MIDlet lifecycle.

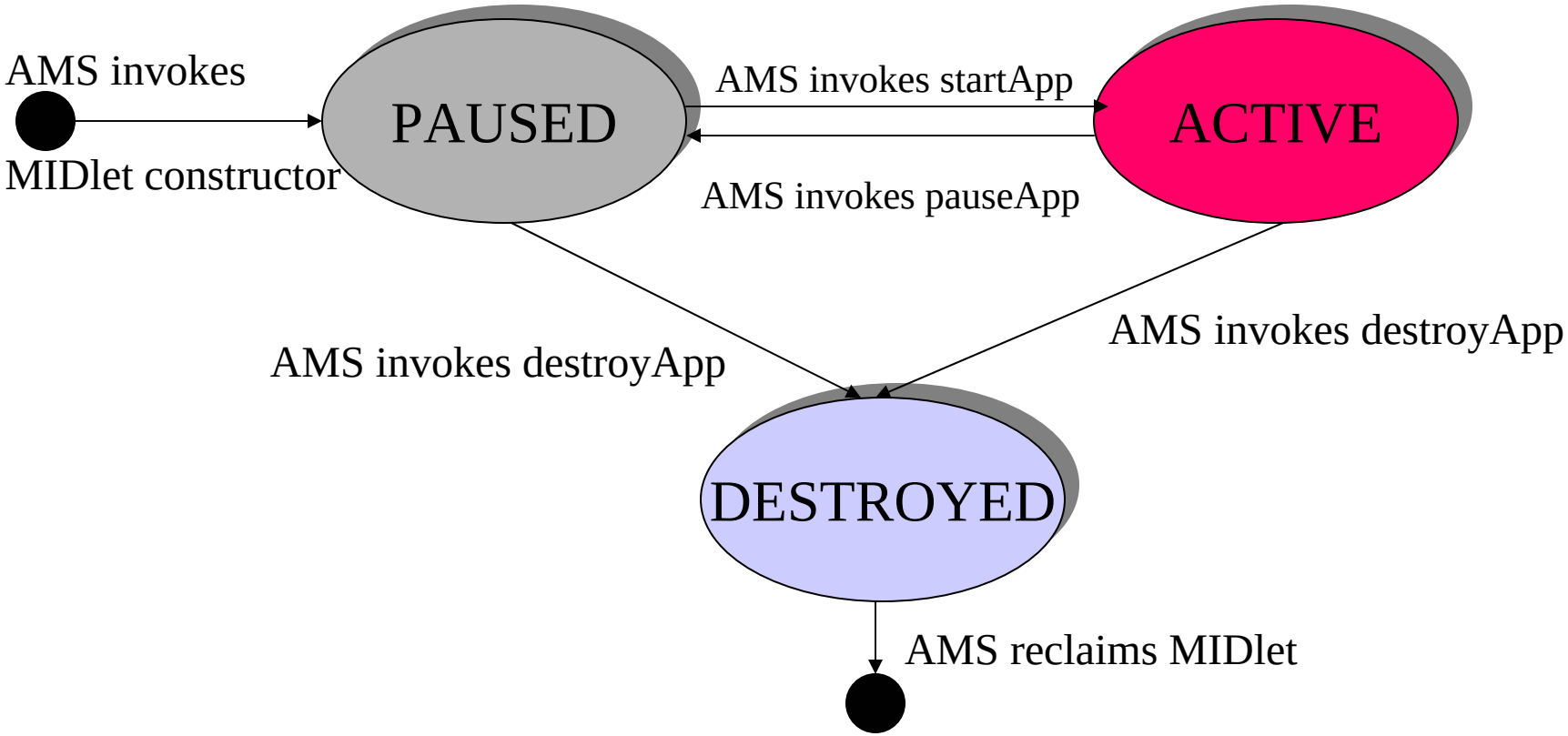
The MIDP Lifecycle

- Every application extends the MIDlet class of the javax.microedition.midlet package.
- The AMS is part of the device's operating environment and guides the MIDlet through its various states during the execution process.
- The MIDlet does not have a public static void main () method.
- MIDlets are initialized when the AMS provides the initial class needed by CLDC to start the MIDlet.

The MIDP States

- PAUSED – the MIDlet has been initialized, but is in a dormant state; when a MIDlet is paused, it should generally release any shared resources.
- ACTIVE – the MIDlet is running; this state is entered after the AMS has called the startApp() method. This method can be called more than once during the MIDlet lifecycle.
- DESTROYED – the MIDlet has released all resources and terminated.

The Lifecycle Model



Notifying and Requesting the AMS

- notifyDestroyed() – notifies AMS that it has released all resources and moved into the DESTROYED state;
- notifyPaused() – notifies AMS that it has released shared resources and moved into the PAUSED state;
- resumeRequest() – a paused MIDlet asks AMS to be resumed;
- getAppProperty() – retrieves named properties from AMS.

The Security Model

- Concepts:
  - trusted MIDlet suites: origin and integrity can be trusted on the basis of some objective criterion;
  - protected APIs: APIs to which access is restricted; the level of access is determined by permissions (Allowed or User) allocated to the API.
- An installed MIDlet suite is bound to a protection domain. A protection domain defines a set of permissions which grant access to an associated set of protected APIs.
- A MIDP 2.0 device must support at least one protection domain, the untrusted domain.
- The set of protection domains supported by an implementation defines the security policy.
- If installed, an unsigned MIDlet suite is always bound to the untrusted domain, in which *access to protected APIs is either denied or requires explicit user permission.*

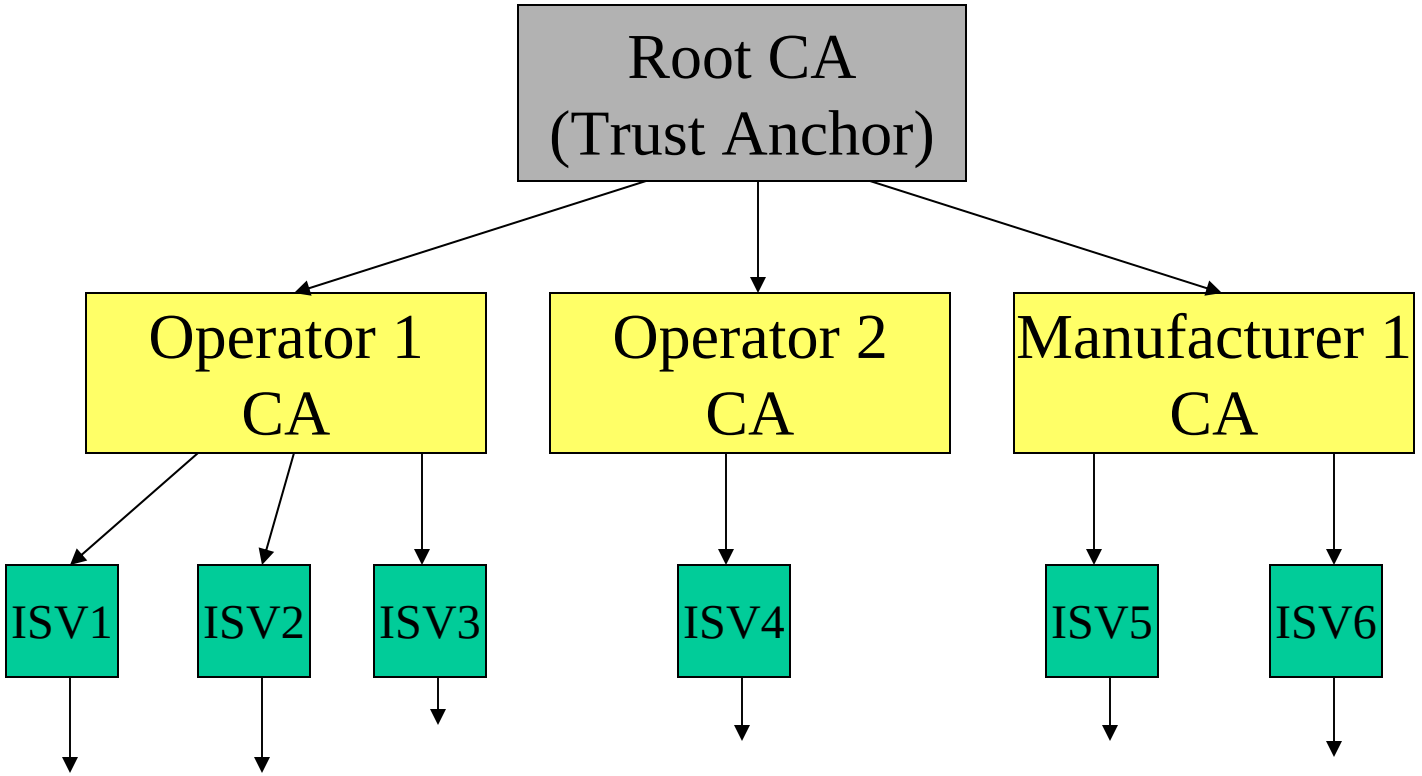
Trusted MIDlet Suites

- MIDP 2.0 does not provide the mechanism for identifying and verifying that a signed MIDlet suite should be bound to a trusted domain; this is left to the device manufacturer and network operators (mobile phones).
- However, MIDP 2.0 defines how the Public Key Infrastructure can be used to identify and verify a signed MIDlet suite.
- The PKI is a system for managing the creation and distribution of digital certificates.
- The public & private keys have two principal uses:
  - secure communication using cryptography,
  - authentication using digital signatures.

The Public Key Infrastructure

- The Public Key Infrastructure: a digest of the document is created which is then encrypted by the private key of the sender. The receiver will use the public key of the sender,…
- The public key is distributed by a trusted certificate authority (CA), as part of a certificate.
- The certificate is digitally signed by the CA using its private key. How can this signature be verified ?
- Root certificates (or root keys) contain details of the CAs and their public keys and are self-signed. They are shipped with the device or embedded in the WIM/SIM card by the network operator (phones). Each certificate will be associated with a trusted protected domain, so that a signed MIDlet that is authenticated against a certificate will be bound to the protection domain associated with that certificate.
- The PKI allows for a hierarchy of CAs up to the trust anchor.

Trust chain



ISV – independent software vendors

Authentication of MIDlet Suites

- The JAR file is signed using the RSA-SHA1 algorithm. The signature in encoded in BASE64 format and inserted into the application descriptor:
- MIDlet-JAR-RSA-SHA1: <base64 encoding of JAR signature>
- The certificate is incorporated into the MIDlet JAD file. In the case of a certification path, all the necessary certificates must be included.
- Before the MIDlet suite is installed, the AMS checks for the presence of the MIDlet-JAR-RSA-SHA1 attribute and will try to authenticate the JAR file.

One certification path

JAD:

MIDlet-Certificate-1-1:<base64 encoding of Supplier's certificate>

MIDlet-Certificate-1-1:<base64 encoding of CA 2's certificate>

MIDlet-Certificate-1-1:<base64 encoding of CA 1's certificate>

- The root certification authority's certificate is available on the device. Using it, the CA 1's public key can be validated.
- This is then used to validate CA 2's public key, which is then used to validate the supplier's public key.
- The supplier's public key is then used to verify the origin and integrity of the JAR file.

Access

- A signed MIDlet suite which includes MIDlets which need access to protected APIs must explicitly request permissions:

MIDlet Permissions: javax.microedition.io.Connector.http,
javax.microedition.io.Connector.https.

- The MIDlet-Permissions attribute appear in the JAD file.
- A MIDlet suite installed as trusted will not be granted any permission it has not explicitly requested.
- The permissions need to be recognized by the device which may grant them, or potentially granted in the protection domain to which the MIDlet suite would be bound.

Protection domains

- A protection domain is a set of permissions determining access to protected APIs or functions.
- Protection domains: Allowed or User (blanket – has permission as installed, session – user authorization is requested the first time the API is invoked, oneshot – user authorization is requested each time the API is invoked).
- The protection domains are defined in a security policy file.

Untrusted MIDlets

- An unsigned MIDlet suite is untrusted – it is bound to the untrusted protection domain.
- They have unrestricted access to:

  javax.microedition.rms
  javax.microedition.midlet
  javax.microedition.lcdui
  javax.microedition.lcdui.game
  javax.microedition.media
  javax.microedition.media.control

- For all the others, they either don't have access or require user authorization.

Recommended security policy

- MIDP 2.0 specification adds to the untrusted domain a set of three protected domains:
  - Manufacturer
  - Operator
  - Third party

- For a trusted domain to be enabled, there must be a certificate identified as a trust root for MIDlet suites in that domain.
- The permissions policy is defined in terms of groups, e.g net_access (http, https, datagram connection, socket, secure socket). For each group, access is either Allowed or User.