


Programación de Dispositivos Móviles




Sesión 19:
Seguridad y activación *push*

Java y Dispositivos Móviles

© 2007-2009 Depto. Ciencia Computación e IA

Seguridad y activación push-1

Índice




- Registro push
- Seguridad

Java y Dispositivos Móviles

© 2007-2009 Depto. Ciencia Computación e IA

Seguridad y activación push-2

Aplicaciones corporativas



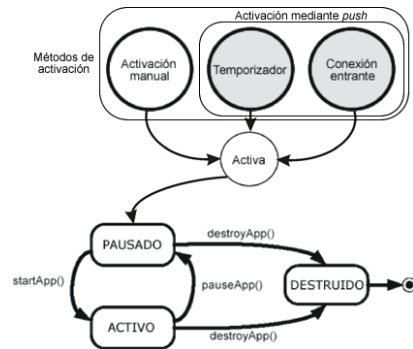
- Registro push
- Seguridad

Java y Dispositivos Móviles

© 2007-2009 Depto. Ciencia Computación e IA

Seguridad y activación push-3

Activación por push



Conexiones entrantes



- Podemos hacer que la aplicación se active cuando se produzca una conexión entrante
 - Sockets, datagramas, mensajes, bluetooth
- Normalmente en el móvil no tendremos una IP fija, por lo que los sockets y los datagramas no son adecuados
- Podemos registrar la conexión push de dos formas:
 - Estática, en el fichero JAD

```
MIDlet-Push-1: sms://:4444,  
es.ua.j2ee.sms.MIDletRecibirSMS, *
```

- Dinámica, utilizando la API de PushRegistry

```
PushRegistry.registerConnection(url, nombreClaseMIDlet,  
remitentesPermitidos);
```

Temporizadores



- Podemos hacer que la aplicación se active a una determinada hora
 - Registraremos un temporizador push con
- ```
long t = PushRegistry.registerAlarm(
midlet.getClass().getName(), fecha.getTime());
```
- Sólo podemos registrar un temporizador push
  - La aplicación no tendrá constancia de que se ha activado mediante push
    - Podemos guardar en RMS información sobre la hora del temporizador
    - Si la aplicación se activa a esta hora, consideramos que ha sido mediante push
  - Las conexiones push sólo serán efectivas cuando nuestra aplicación esté cerrada
    - Cuando esté abierta será responsabilidad de nuestro MIDlet responder a los temporizadores y a la conexiones entrantes

---

---

---

---

---

---

---

---

## Aplicaciones corporativas



- Registro push
- Seguridad

---

---

---

---

---

---

---

---

## Seguridad



- Para garantizar la seguridad, las aplicaciones MIDP se ejecutan dentro de una caja de arena (*sandbox*)
  - Entorno restringido
  - Evitar que causen daños a otras aplicaciones del dispositivo
- Están restringidas a acceder únicamente a los recursos de la *suite* de la aplicación
  - Sólo puede utilizar clases Java de su propia suite
  - Sólo puede leer recursos estáticos contenido dentro de su suite
  - Sólo puede acceder a almacenes de registros creados por MIDlets de su misma suite
  - No pueden acceder al sistema de ficheros del móvil, si necesitan almacenar datos debe hacerlo mediante RMS
  - Sólo pueden usar la API Java (MIDP), nunca a la API nativa

---

---

---

---

---

---

---

---

## Seguridad en MIDP 2.0



- Operaciones sensibles
  - Establecer conexiones de red
  - Registrar activación por push
- Debemos solicitar permiso para realizar estas operaciones
  - Lo haremos en el fichero JAD mediante el atributo `MIDlet-Permissions`
- Al instalar la aplicación se asigna a un dominio
  - Según el dominio se concederán ciertos permisos
- El dispositivo asignará dominios que otorguen permisos a las aplicaciones que sean de confianza

---

---

---

---

---

---

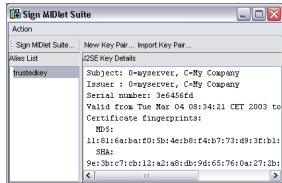
---

---

## Firmar aplicaciones



- Se recomienda que se utilicen firmas y certificados para decidir el dominio que se le asignará a cada aplicación
- Cada dispositivo tendrá almacenado un conjunto de firmas
  - Deberemos utilizar una de estas firmas para que nuestra aplicación sea de confianza
- Podemos utilizar WTK para realizar pruebas con MIDlets firmados



---

---

---

---

---

---

---

---