

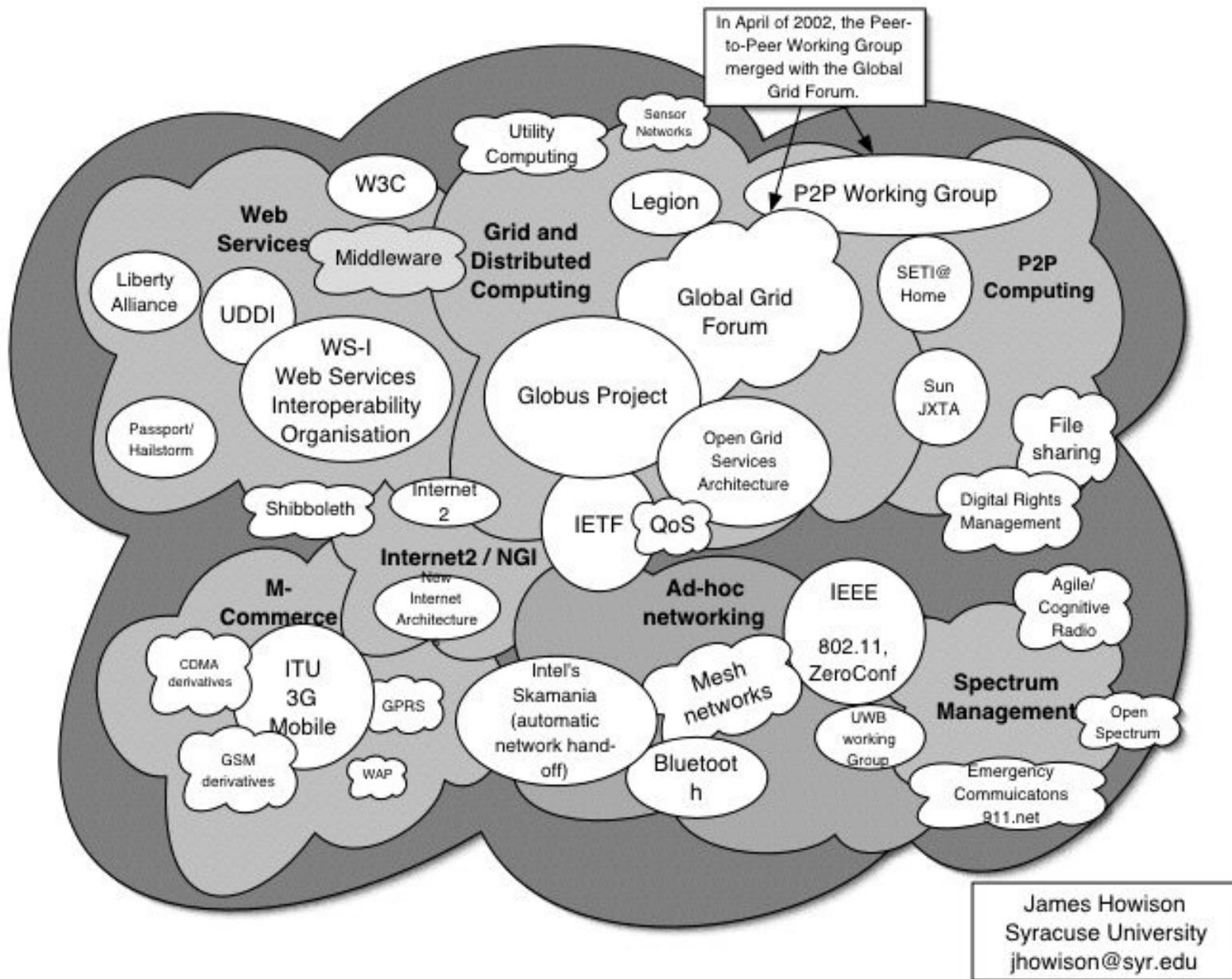
Wireless Grids e Pervasive Grids

Wireless Grids e Pervasive Grids

- **Wireless Grids**
 - Caratteristiche
 - Sistemi
 - Applicazioni
- **Pervasive Grids**
 - Caratteristiche
 - Problemi e strategie implementative

Wireless Grids

- Le Grid oggi sono essenzialmente Wired Grids con nodi di granularità medio/grande.
- Le nuove tecnologie wireless offrono soluzioni tecnologiche per realizzare Wireless Grids e Pervasive Grids composte da un numero elevato di nodi di granularità medio/piccola.
- Nuove architetture, nuove soluzioni, nuove applicazioni e nuove sfide.



Wireless Grids

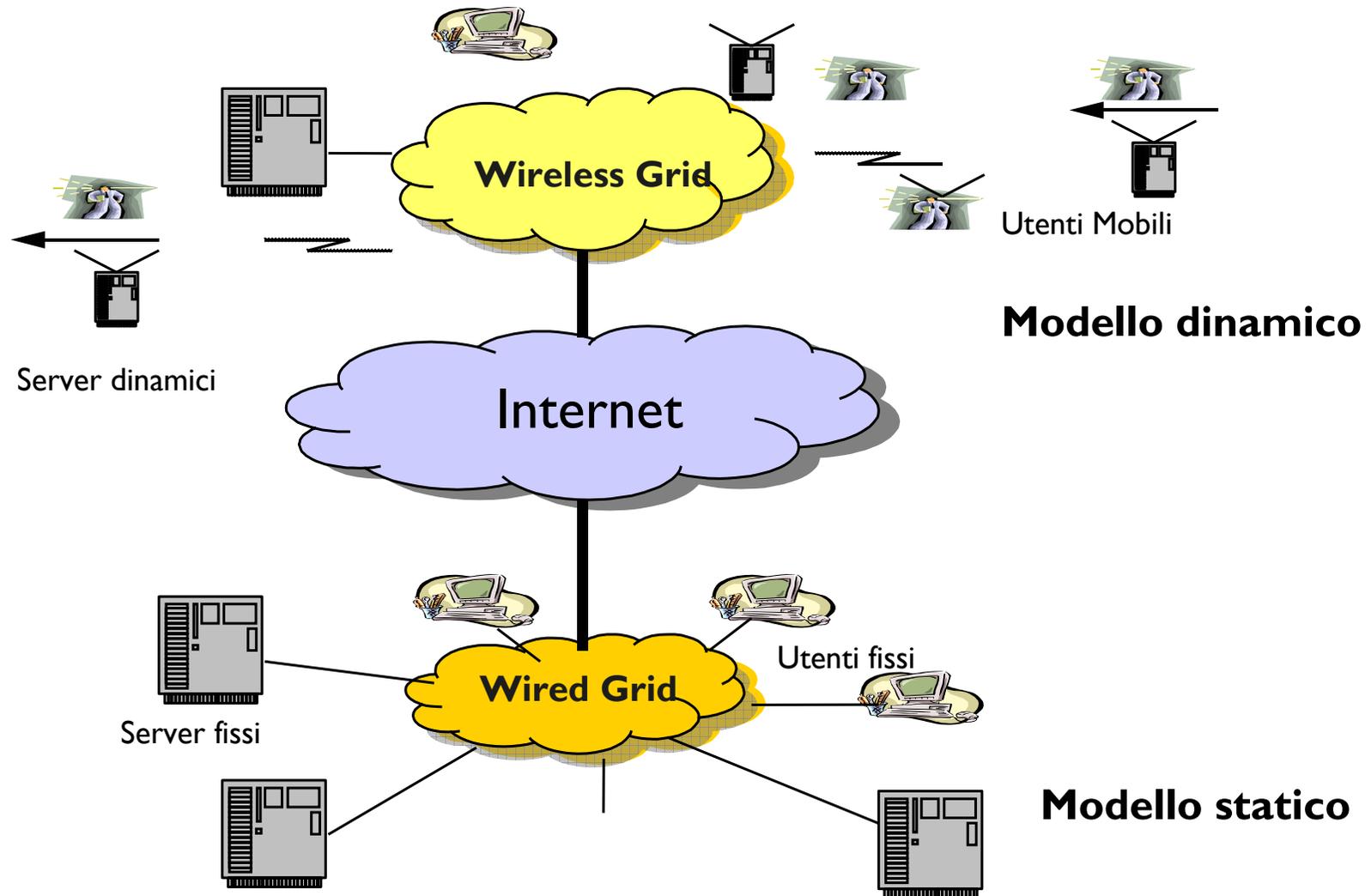
- Le Wireless Grids possono essere viste come:
 - Una Front-End/User Interface per le Wired Grids
 - Una rete da usare per condividere risorse mobili
 - Una Grid di sensori a bassa potenza connessi tra loro
 - Integrazione di Wifi, Bluetooth, smart phones con Grid middleware e Grid Services
 - Una Griglia oltre la stratosfera → **Galaxy Grid**



Caratteristiche di una Wireless Grid

- Dispositivi piccoli e a basso consumo
 - Problemi di alimentazione
- Dispositivi mobili e nomadic
 - Cellulari, PDA, laptop, e RFID
 - Car Grids
- Aggregazione e auto-organizzazione di small computing elements
 - Resource pooling
- Reti di sensori wireless
 - Smartdust, or a massive mobile phone computing/commun.
 - Risk management,

Wireless Grids



Classi di Applicazioni

- Applicazioni che aggregano informazioni che arrivano da dispositivi mobili/wireless: data warehouse, data compression, data aggregation, ...
- Applicazioni che sfruttano informazioni che arrivano da dispositivi presenti in località remote: data streaming, sensor grids, ...
- Applicazioni che sfruttano la capacità di auto-organizzazione di dispositivi pervasivi.

Protocolli e Servizi

- Descrizione delle Risorse (WSDL e XML)
- Discovery delle Risorse (UDDI, Index services)
- Comunicazione: one-to-one, IP multicast, ...
- Coordinamento e auto-organizzazione
- Trusting e security
- Fault-tolerant

Middleware per Wireless Grids

- Globus 4 – Open Grid Service Architecture
 - SOAP e XML
 - WSRF
- Power Efficient Routing
 - Protocolli power-efficient per applicazioni Wireless Grid
- Verso Grid Services “leggeri”
 - WSRF services in ambienti con risorse “scarse”
 - Protocolli basati su aggregazione

Web Services in ambienti mobili

- I Web Services possono essere utilizzati in ambienti mobili secondo tre modelli architetturali:
 - I. **Wireless portal network:** viene utilizzato un *gateway* tra il client mobile ed il Web Service
 1. il client invia le richieste al gateway utilizzando un formato specifico (più leggero di SOAP e quindi più adatto a trasmissioni wireless su banda limitata);
 2. il gateway traduce le richieste in messaggi SOAP e li invia al Web Service;
 3. il Web Service invia la risposta al gateway utilizzando SOAP;
 4. il gateway restituisce la risposta al client utilizzando il formato specifico.

Web Services in ambienti mobili

- 2. Wireless extended Internet:** i client mobili interagiscono direttamente con i Web Services. In questo caso i client mobili agiscono a tutti gli effetti come client di Web Services e possono inviare e ricevere messaggi SOAP.
- 3. Peer-to-Peer (P2P) network:** i dispositivi mobili possono agire sia da client sia da fornitori di Web Services. Non è attualmente implementato in sistemi reali, ma potrebbe affermarsi nel prossimo futuro.

La libreria JSR-172 per J2ME

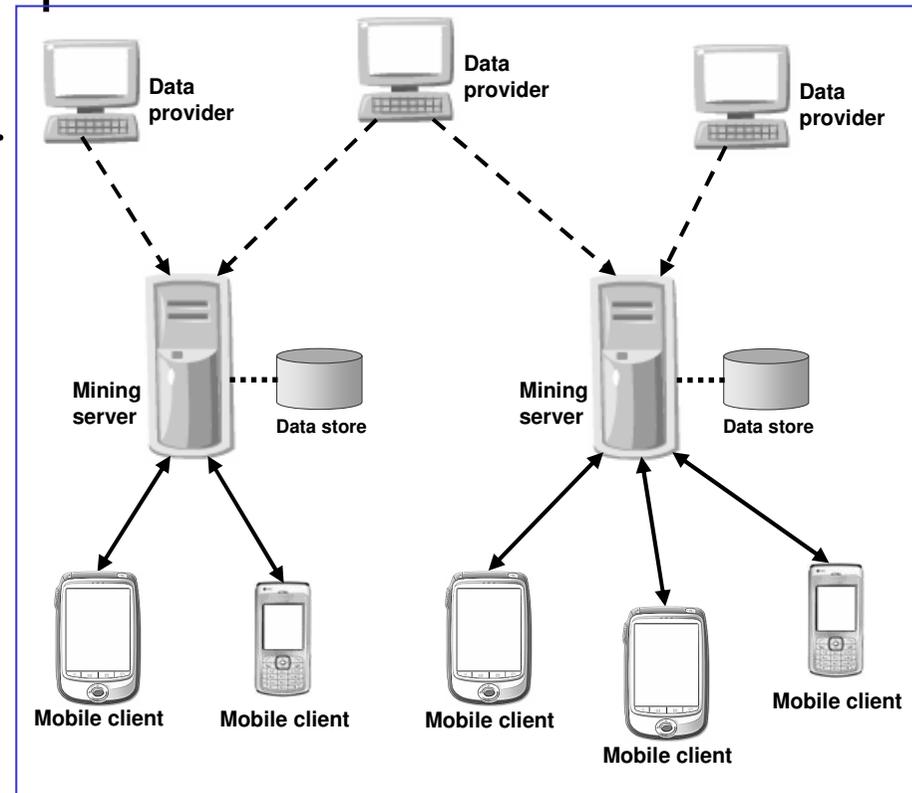
- Il modello “Wireless extended Internet” si sta affermando negli ultimi anni grazie ad una libreria aggiuntiva per J2ME, denominata **JSR-172**.
- JSR-172 consente ad un client J2ME di invocare direttamente un Web Service remoto.
- Esempio:
un sistema che fa uso di J2ME e della libreria JSR-172 per l’accesso a Web Services da terminali mobili (cellulari, PDA, etc.).

La libreria JSR-172 per J2ME

- JSR-172 include un insieme di librerie specifiche, tra cui:
 - librerie per la manipolazione di dati XML, basate su un sottinsieme delle librerie standard per il parsing di documenti XML;
 - librerie per abilitare la comunicazione remota basata su XML, inclusi strumenti per la generazione degli *stub* a partire dal WSDL del servizio.

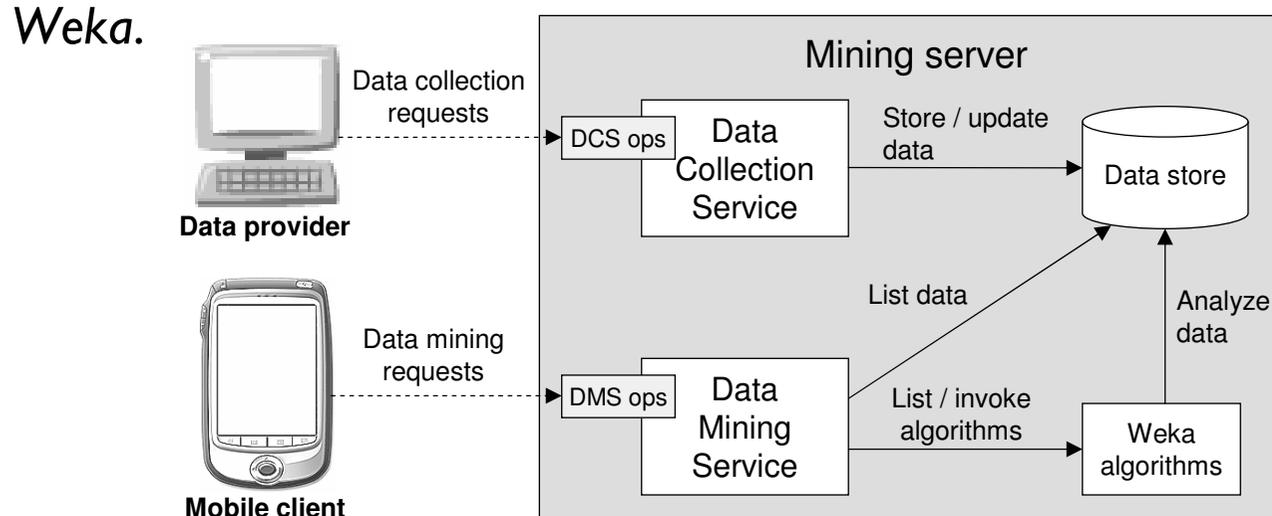
Use Case: Web Services per mobile data mining

- L'obiettivo del sistema è supportare l'accesso a servizi di *data mining* da dispositivi mobili.
- Il sistema include tre tipi di componenti:
 - **Data provider:** le applicazioni che generano i dati da analizzare.
 - **Mining server:** nodi server utilizzati per memorizzare i dati generati dai data providers e per eseguire i task di data mining sottomessi dai mobile client.
 - **Mobile client:** le applicazioni che richiedono l'esecuzione di task di data mining sui dati remoti.



Mining server (I)

- I Mining server forniscono le loro funzionalità mediante due Web Services:
 - **Data Collection Service (DCS):** invocati dai data provider per memorizzare dati nel *data store*.
 - **Data Mining Service (DMS):** invocati dai mobile client per eseguire task di data mining
 - l'analisi dei dati è eseguita utilizzando gli algoritmi della libreria *Weka*.



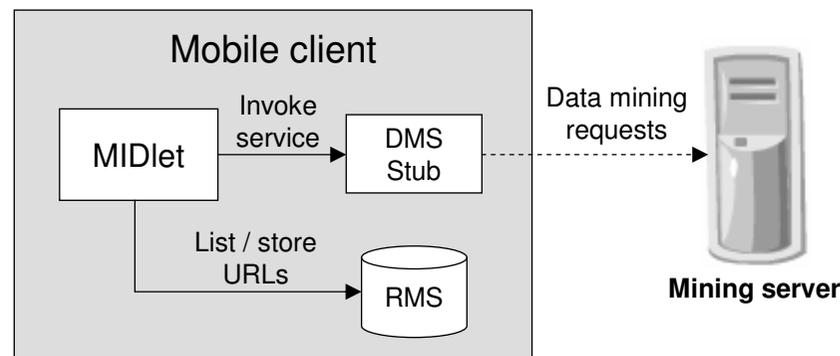
Mining server (2)

- Operazioni implementate dal Data Mining Service (DMS):

Operazione	Descrizione
listDatasets	Restituisce la lista dei dataset locali.
listAlgorithms	Restituisce la lista degli algoritmi di data mining disponibili.
submitTask	Sottomette un task di data mining per l'analisi di un dato dataset usando uno specifico algoritmo di data mining. Restituisce un <i>id</i> univoco per il task.
getStatus	Restituisce lo stato corrente del task con un dato <i>id</i> . Lo stato può essere <i>running</i> , <i>done</i> , o <i>failed</i> .
getResult	Restituisce il risultato del task con un dato <i>id</i> , in formato testuale o visuale.

Mobile client

- Il client mobile è composto da tre componenti:
 - **MIDlet**: l'applicazione J2ME che consente all'utente di eseguire operazioni di data mining e visualizzare i risultati.
 - **DMS Stub**: lo stub che consente alla MIDlet di invocare le operazioni di un Data Mining Service (DMS) remoto.
 - **Record Management System (RMS)**: un database record-oriented che consente alle applicazioni J2ME di memorizzare dati tra un'invocazione e l'altra (utilizzata per memorizzare gli URL dei DMS disponibili).



Ciclo di vita dell'applicazione (I)

1. L'utente avvia la MIDlet. La MIDlet accede all'RMS e prende la lista dei mining server remoti. La lista è mostrata all'utente, il quale sceglie il mining server da connettere.
2. La MIDlet invoca le operazioni **listDatasets** e **listAlgorithms** dei DMS remoti per ricevere la lista dei dataset e degli algorithmi disponibili sul server. Le liste sono mostrate all'utente che sceglie il dataset da analizzare e l'algoritmo da utilizzare.
3. La MIDlet invoca l'operazione **submitTask** del DMS remoto, passando il dataset e l'algoritmo selezionato dall'utente. Il task è sottomesso in modalità batch: subito dopo che il task è stato sottomesso, il DMS restituisce un *id* univoco, e la connessione tra client e server viene rilasciata.

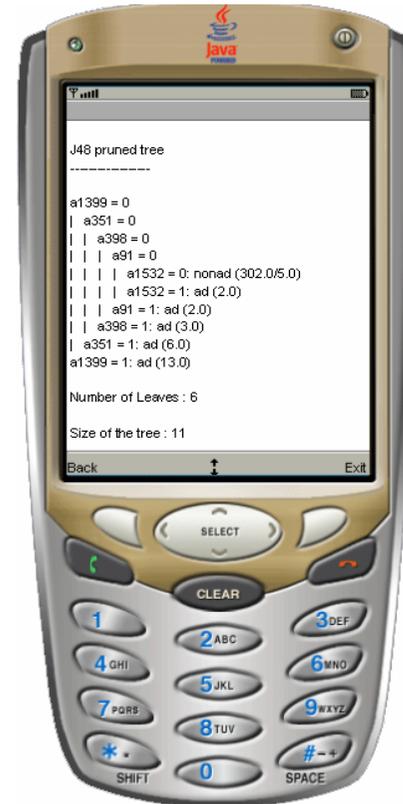
Ciclo di vita dell'applicazione (2)

4. Dopo la sottomissione del task, la MIDlet monitora il suo stato interrogando il DMS. A tal fine, la MIDlet invoca periodicamente l'operazione **getStatus**, che riceve l'*id* del task e restituisce il suo stato corrente (vedi tabella). L'intervallo di polling può essere definito dall'utente.
5. Appena l'operazione **getStatus** restituisce *done*, la MIDlet invoca l'operazione **getResult** per ricevere il risultato del task di data mining. In base al tipo di task, la MIDlet chiede all'utente di scegliere come visualizzare il risultato della computazione (per es.: pruned tree, matrice di confusione, etc.).

Interfaccia utente (MIDlet)



Selezione del risultato da visualizzare



Visualizzazione del risultato

Security e Trust

- End-to-end security e trust
- Content Monitoring vs Privacy
 - Uno o l'altro
 - Approccio integrato
 - ...

Modelli di Business

- **Grid Market:**
 - Open vs Closed model
 - Chi guadagna per cosa ?

- **Condivisione di risorse e informazione**
 - Chi paga per cosa?
 - Come si paga?
 - Grid Collaboration e cooperative markets

Pervasive Grids

- Modelli futuri di computazione distribuita
 - Integrazione di elementi di elaborazione e comunicazione a “grana” diversa
- Comunicazione/sincronizzazione tra sensori, PDA, PC, clusters, RFID, wearable computers.
- Verso una WWG pervasiva.

Conclusioni

- Tecnologia hardware già disponibile
- Tecnologie di comunicazione in forte sviluppo
- Protocolli, middleware e servizi in stato preliminare
- Futuri modelli, protocolli e architetture
- Oggi è comunque possibile realizzare wireless Grids e pervasive Grids.