

# Java ME e Web 2.0

**nuovi scenari applicativi per dispositivi mobili**

Stefano Sanna

*gerdavax@tiscali.it - <http://www.gerdavax.it>*

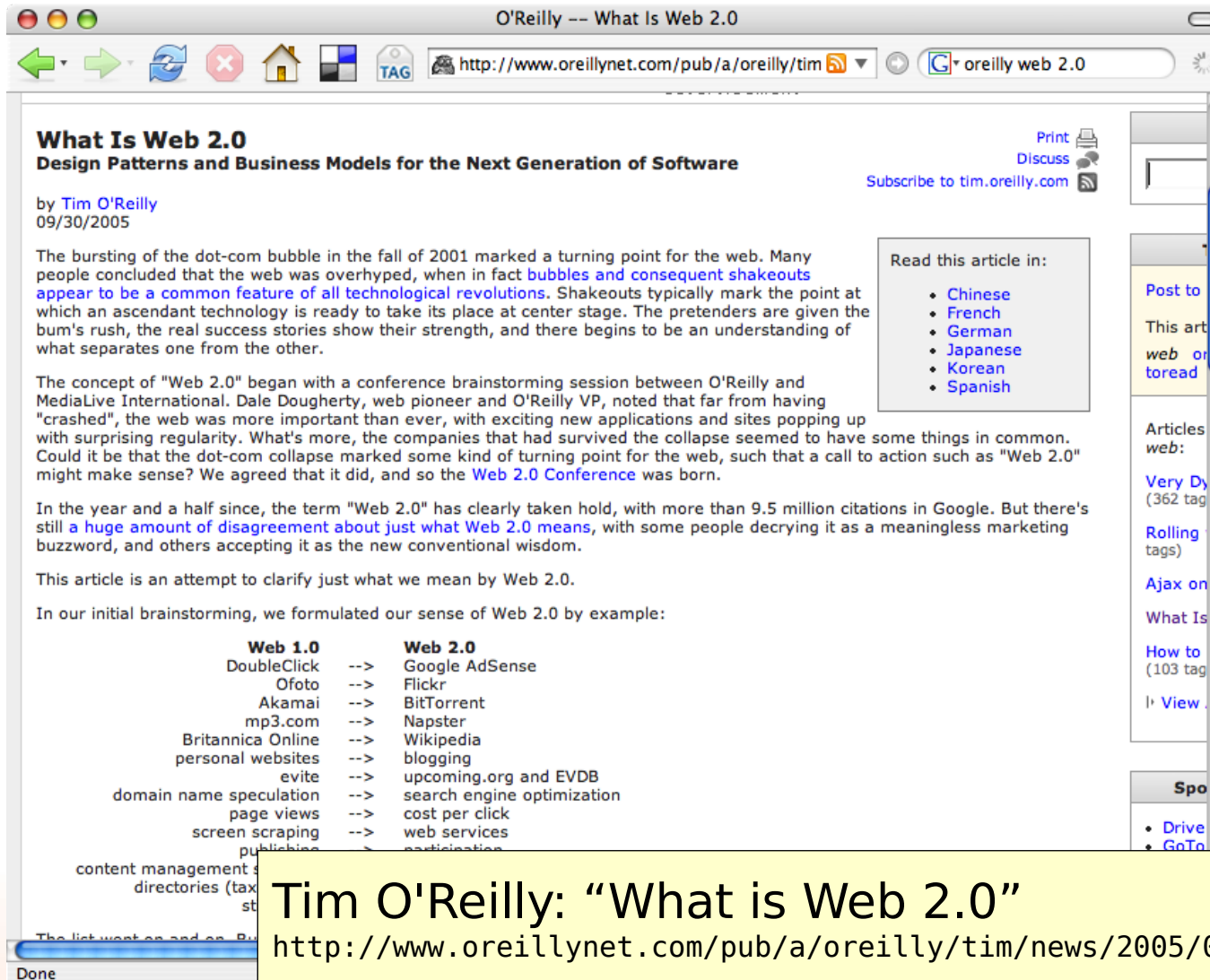
*Java Italian Association*

*<http://www.jia.it>*

- Il contesto:
  - Web 2.0: who's who
  - Java ME: a che punto siamo?
  - Mobile interface per servizi Web 2.0
- Due esempi pratici:
  - Bloglines
  - Upcoming (e query georeferenziate)
- Conclusioni e riferimenti

- L'evoluzione che conduce a considerare il **Web come una piattaforma** per erogare servizi e non più un strumento di comunicazione
- *Buzzword* con cui si identifica una nuova classe di applicazioni, di strumenti di sviluppo, di servizi... di **opportunità nel settore ICT**
- Si potrebbe definire per enumerazione:
  - AJAX, Comet, RSS, Atom, REST...
  - Gmail, Google Maps, Posticky, del.icio.us, Flickr, Bloglines, Blogger, Technorati...

# Web 2.0: cosa è?



**What Is Web 2.0**  
Design Patterns and Business Models for the Next Generation of Software

by Tim O'Reilly  
09/30/2005

The bursting of the dot-com bubble in the fall of 2001 marked a turning point for the web. Many people concluded that the web was overhyped, when in fact [bubbles and consequent shakeouts appear to be a common feature of all technological revolutions](#). Shakeouts typically mark the point at which an ascendant technology is ready to take its place at center stage. The pretenders are given the bum's rush, the real success stories show their strength, and there begins to be an understanding of what separates one from the other.

The concept of "Web 2.0" began with a conference brainstorming session between O'Reilly and MediaLive International. Dale Dougherty, web pioneer and O'Reilly VP, noted that far from having "crashed", the web was more important than ever, with exciting new applications and sites popping up with surprising regularity. What's more, the companies that had survived the collapse seemed to have some things in common. Could it be that the dot-com collapse marked some kind of turning point for the web, such that a call to action such as "Web 2.0" might make sense? We agreed that it did, and so the [Web 2.0 Conference](#) was born.

In the year and a half since, the term "Web 2.0" has clearly taken hold, with more than 9.5 million citations in Google. But there's still [a huge amount of disagreement about just what Web 2.0 means](#), with some people decrying it as a meaningless marketing buzzword, and others accepting it as the new conventional wisdom.

This article is an attempt to clarify just what we mean by Web 2.0.

In our initial brainstorming, we formulated our sense of Web 2.0 by example:

Web 1.0	Web 2.0
DoubleClick	--> Google AdSense
Ofoto	--> Flickr
Akamai	--> BitTorrent
mp3.com	--> Napster
Britannica Online	--> Wikipedia
personal websites	--> blogging
evite	--> upcoming.org and EVDB
domain name speculation	--> search engine optimization
page views	--> cost per click
screen scraping	--> web services
publishing	--> participation
content management	
directories (taxi)	
st	

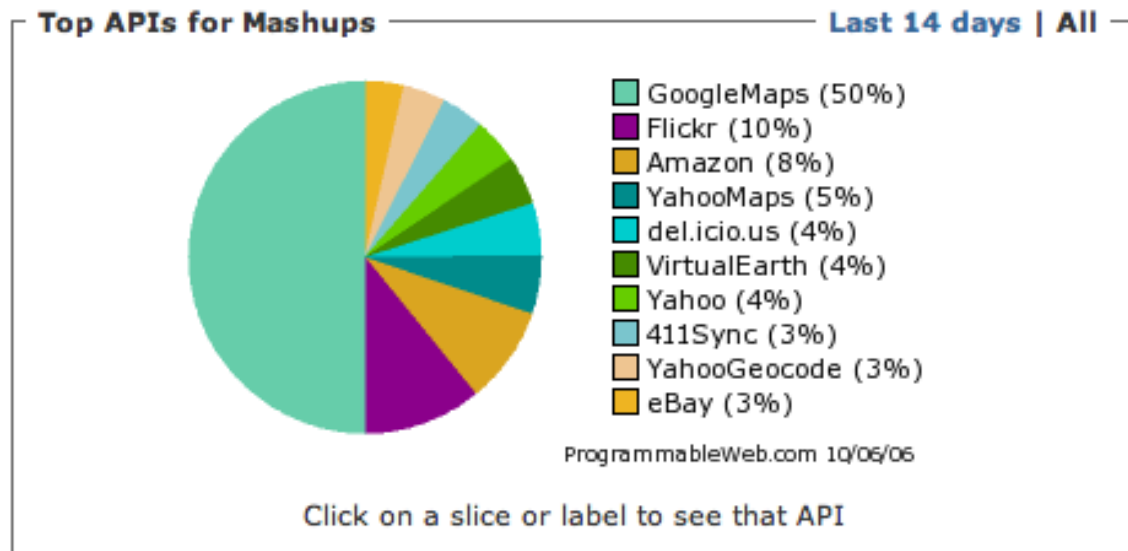
The list went on and on. But

Done

Tim O'Reilly: "What is Web 2.0"  
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

- Nuovo approccio tecnologico
  - rapido (perpetual beta!), efficace, efficiente
  - servizi aperti e **accessibili con protocolli semplici**
  - possibilità di estendere e comporre servizi esistenti (mashup)
- Nuovo approccio sociale
  - fiducia e collaborazione, prima di tutto!
  - massima apertura verso **soluzioni custom**
  - il servizio sopravvive se gli utenti lo utilizzano, lo modellano (hacking) e lo popolano di dati

- Il numero di servizi, API, mashup disponibili è in continua crescita!
- Il sito programmableweb.com riporta al 6/10:
  - **1040** applicazioni mashup rilevate
  - circa 2.6 applicazioni **mashup nuove** ogni giorno
  - **284** API



# Le API più utilizzate

## API Cloud

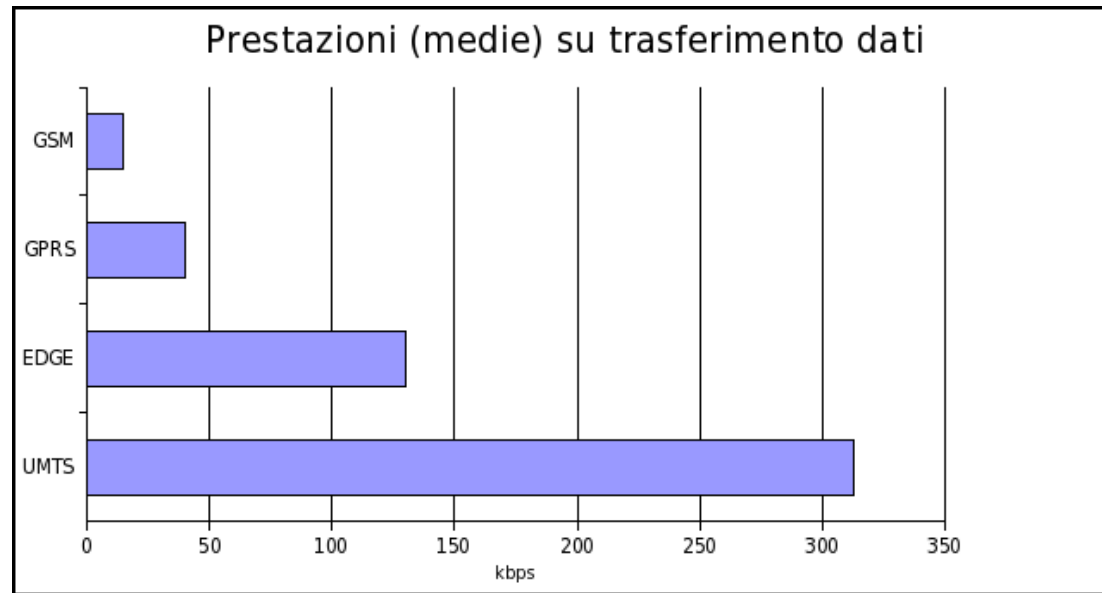
23 30Boxes 411Sync 43Things A9 AIM Alexa AlexaWebInfo  
Amazon AmazonQueue AmazonS3 AOLMusicNow Backpack  
Basecamp BBC Blogger Bloglines Buzznet CafePress Dapper  
del.icio.us DigitalPodcast EasyUtil eBay Entrez eSideWalk ESV  
EVDB Facebook FedEx FeedBurner FeedMap Findory Flickr  
geocoder geocoderCanada GeoNames Gigablast Google  
GoogleAdSense GoogleAdWords GoogleAjaxSearch GoogleBase  
GoogleCalendar GoogleDesktop GoogleHomepage GoogleMaps  
GoogleTalk GrouperVideo hostip HotOrNot Interfax InternetArchive  
Jots JotSpot Kayak LastFM LiveJournal Map24 MapPoint MapQuest  
MechanicalTurk Moreover MSNMessenger MSNSearch  
MusicBrainz NASA Naver NewsGator NOAA Ontok OntokWikipedia  
PayPal Pixagogo Prodigem PubSub RapLeaf RawSugar Rhapsody Riya  
Salesforce SecondLife Shadows Shopping.com Simpy Skype  
Smugmug SplogSpot Start TagTooga Tailrank Technorati Telcontar  
Textamerica Topicalizer Trynt TypeKey TypePad Upcoming UPS  
UrbanDictionary VirtualEarth WeatherBug Weblogs WebShots  
WindowsLiveExpo Yahoo YahooAds YahooAnswers YahooAudio  
YahooGeocode YahooImages YahooLocal  
YahooMapImage YahooMaps YahooRelatedQuery YahooShopping  
YahooTerms YahooTraffic YahooTravel YahooVideo YahooWidgets  
YouTube ZeeMaps ZoomIn Zvents



- In ambito mobile è possibile intravedere una rivoluzione copernicana simile a quanto avvenuto sul Web. In particolare il dispositivo mobile evolve in precise direzioni:

mobile browser	-->	mobile blogger
data provisioning	-->	mobile data provider
network-aware	-->	context-aware
stand-alone	-->	augmented environment
yet-another-device	-->	my constant touch

- L'aumentare della banda disponibile sulle reti wireless è fattore **determinante** per l'accesso a LWS attraverso dispositivo mobile



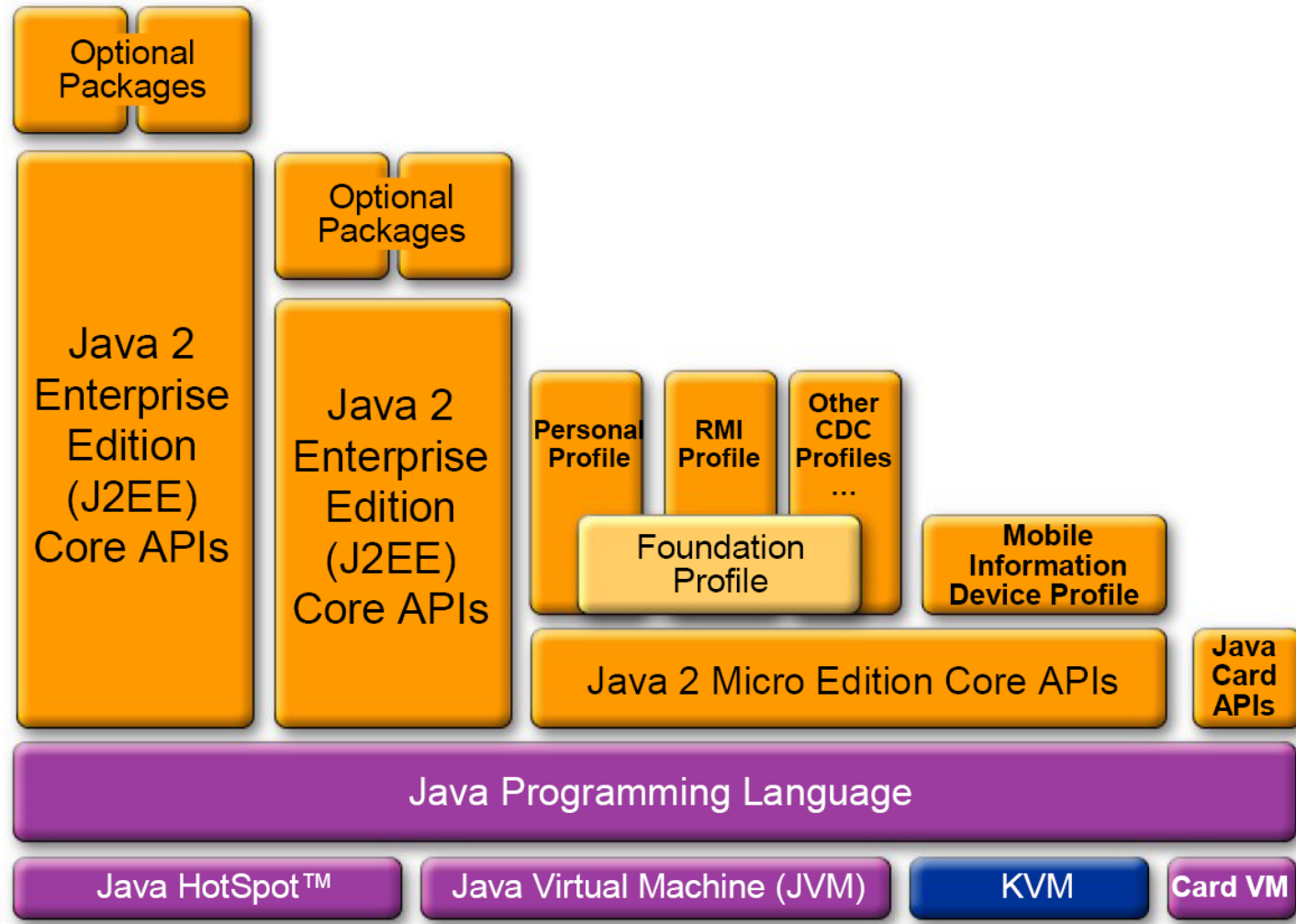
- HSDPA promette velocità sino a 1.8Mbps, ma la copertura è ancora molto limitata

# Java ME: a che punto siamo?

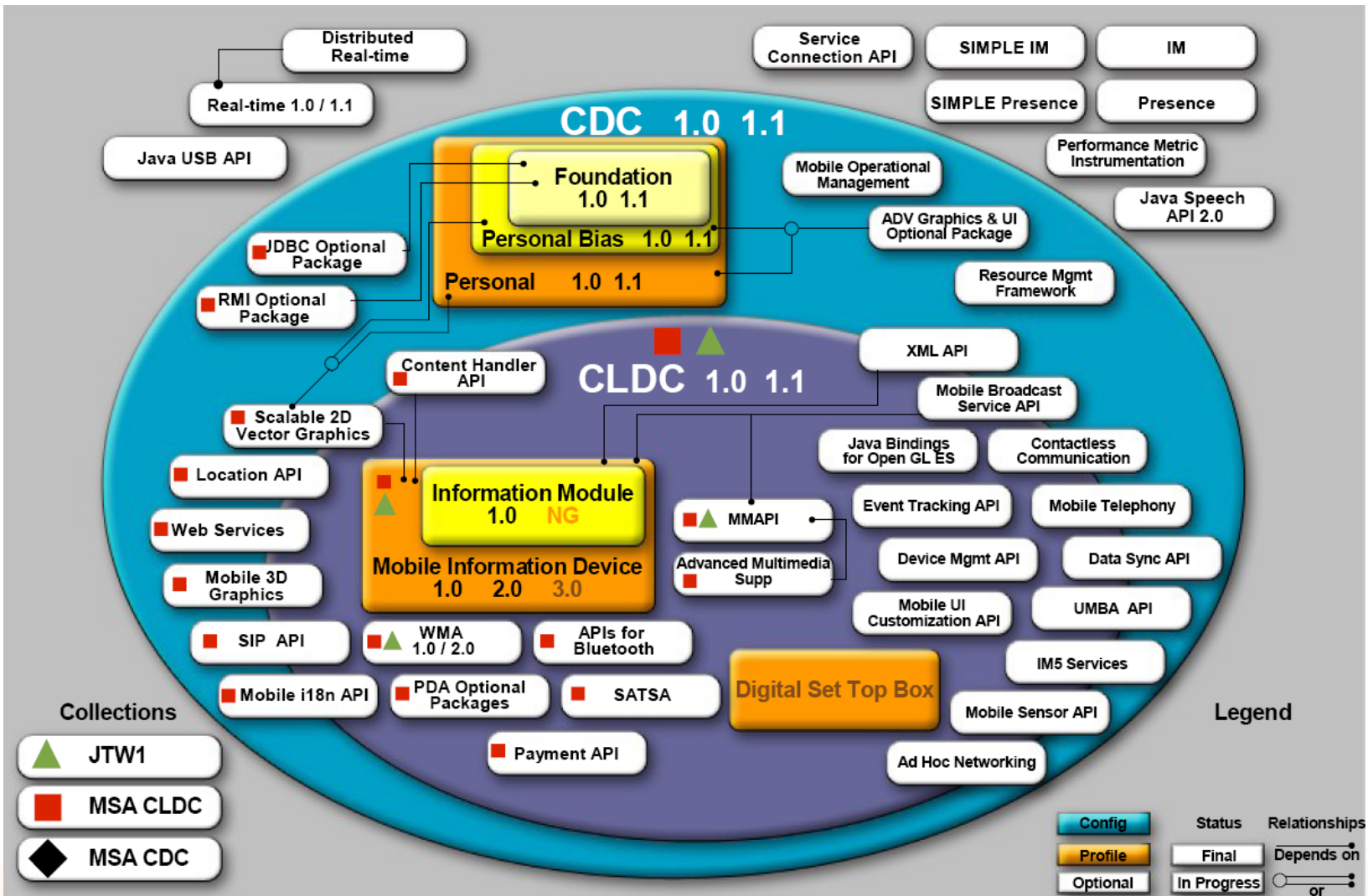
- Timeline:
  - Settembre 2001: MIDP 1.0
  - Novembre 2002: MIDP 2.0
  - fine 2006 / inizio 2007: MIDP 3.0
- Due segnali più importanti dell'imminente rilascio di MIDP 3.0:
  - specifica e implementazione di una marea di API opzionali  
(<http://jcp.org/en/jsr/tech?listBy=1&listByType=platform>)
  - supporto pieno CDC e profili superiori (Nokia Series 60, Windows Mobile, Savaje...)



# Java ME, ieri...



# Java ME, oggi!



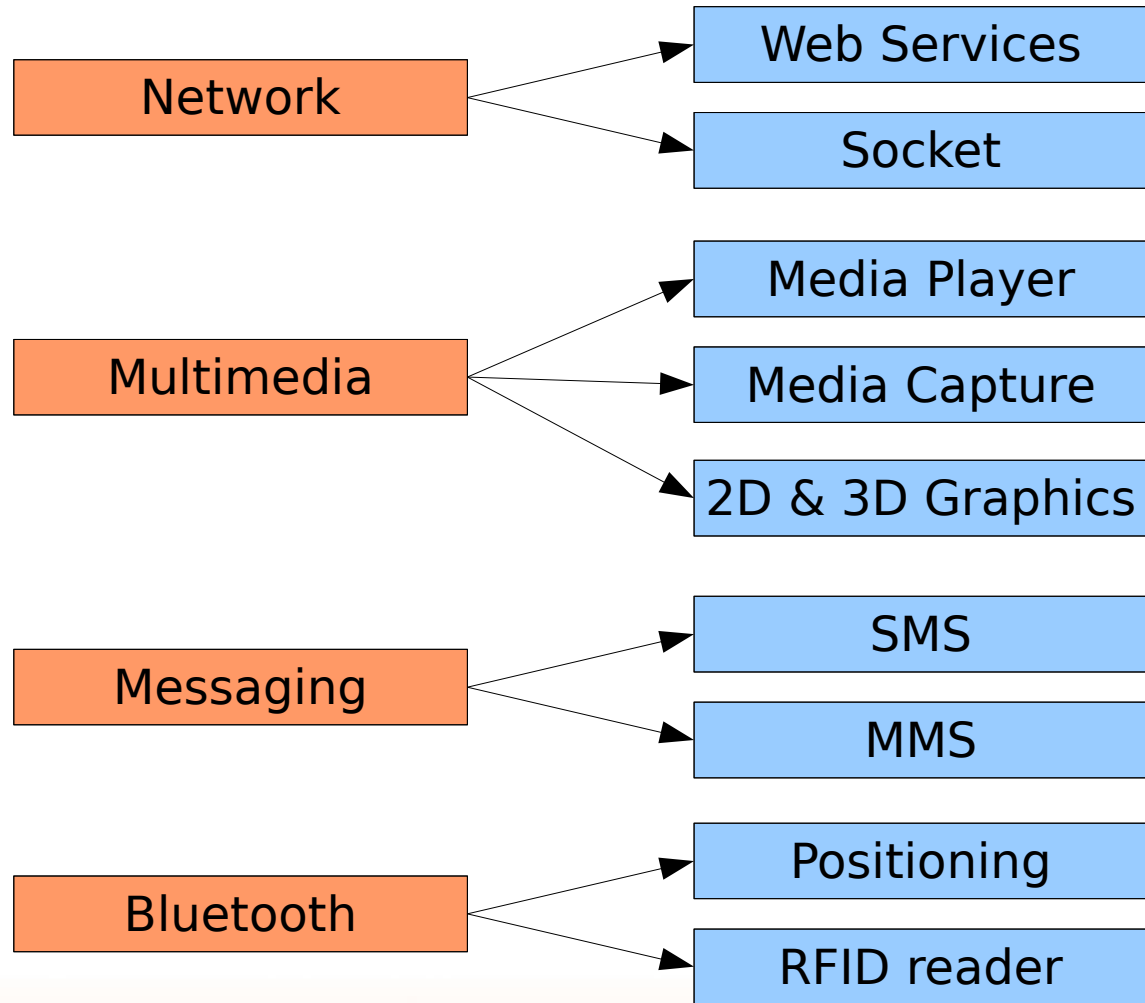
- Le API opzionali sono la vera ricchezza:
  - Connettività:
    - JSR 82: Bluetooth API
    - JSR 172: Web Services API
    - JSR 120 & 205: SMS & MMS API
    - JSR 180: SIP API
  - Multimedia:
    - JSR 135: Mobile Media API
    - JSR 184: Mobile 3D Graphics
    - JSR 206: Scalable 2D Vector Graphics
    - JSR 234: Adv. Multimedia API (camera control)



Nokia N95

- Java ME è la tecnologia abilitante che permette all'utente di **accedere a servizi Web 2.0 in mobilità** utilizzando servizi e protocolli standard
- Sono disponibili tutti i componenti:
  - connettività http/https
  - integrazione con il browser locale
  - persistenza (salvataggio impostazioni, caching, possibilità di operare anche in assenza di connettività)
  - accesso a servizi di messaggistica, Bluetooth, grafica 3D...

# L'offerta di servizi



- Diversi ambienti di sviluppo forniscono strumenti per realizzare applicazioni mobile

- Piattaforme native:

- Symbian OS
- Windows Mobile
- Linux

**symbian**

Symbian OS - the mobile operating system



*maemo.org* 2.0

- Execution environment:

- Python (su Series 60 e Maemo)
- FlashLite
- Mobile Opera Platform



- Quattro approcci possibili:
  - Pre-processing su server remoto e trasferimento su protocollo proprietario
    - soluzione non praticabile al crescere del numero di servizi da supportare
  - XML-RPC
    - non consente la definizione di tipi di dati custom
  - Web Services (SOAP) + local viewer
    - troppo oneroso, compromesso da incompatibilità
  - Lightweight Web Services (REST-like)
    - acquista sempre maggiore interesse da parte dell'industria e degli sviluppatori

- I problemi più frequenti in ambito Java ME:
  - la maggior parte dei web services esistenti (per di più la maggior parte di quelli pubblici!) non è conforme allo standard WS-I Basic Profile
  - talvolta l'encoding richiesto dal servizio non è supportato dal dispositivo
  - i messaggi SOAP comportano maggiore occupazione di banda, memoria e tempo CPU per la decodifica

- I servizi Web 2.0 sono semplici, “atomici”, leggeri e universalmente accessibili
- Poche linee di codice (meglio se con un linguaggio di scripting) permettono di realizzare un client perfettamente funzionante
- Sono l'interfaccia più adatta a supportare l'accesso da dispositivo mobile

- REST: Representational State Transfer
- Interfaccia orientata allo scambio dello stato di oggetti anziché all'invocazione di metodi
- Principi semplici:
  - l'**oggetto** del quale interessa lo stato è **individuato** da una **URL**
    - ad esempio: <http://service.com/catalog/hardware>
  - lo **stato** dell'oggetto è **trasferito** preferibilmente in XML
- Wikipedia:
  - [http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)

# Esempio di servizi REST

- YouTube:
  - [http://www.youtube.com/dev\\_rest](http://www.youtube.com/dev_rest)
- eBay:
  - <http://developer.ebay.com/developercenter/rest>
- Upcoming:
  - <http://upcoming.org/services/api/>
- Technorati:
  - <http://technorati.com/developers/api/>
- **Esplorare per innovare:**
  - <http://www.programmableweb.com/apilist>

- Implementazione molto semplice, anche in assenza di API dedicata
- Traffico dati ridotto al minimo indispensabile
- Nessun vincolo legato alla conformità con la WS-I Basic Profile (si veda la JSR 172)
- I servizi REST, RESTful, REST-like sono in continuo aumento!!!

- Raramente viene fornita DTD o Schema dell'XML utilizzato: il parser deve essere costruito manualmente leggendo i documenti di specifica
- Servizi come quelli elencati **possono cambiare spesso interfaccia**, dunque anche i client devono essere verificati (e aggiornati) costantemente
- Inutilizzabile per architetture complesse

- Consideriamo ora due esempi semplici ma significativi della potenza dei nuovo lightweight web services:
  - **Bloglines**
    - per ricevere notifica degli aggiornamenti dei feed di interesse
  - **Upcoming**
    - per ottenere informazioni su eventi di interesse

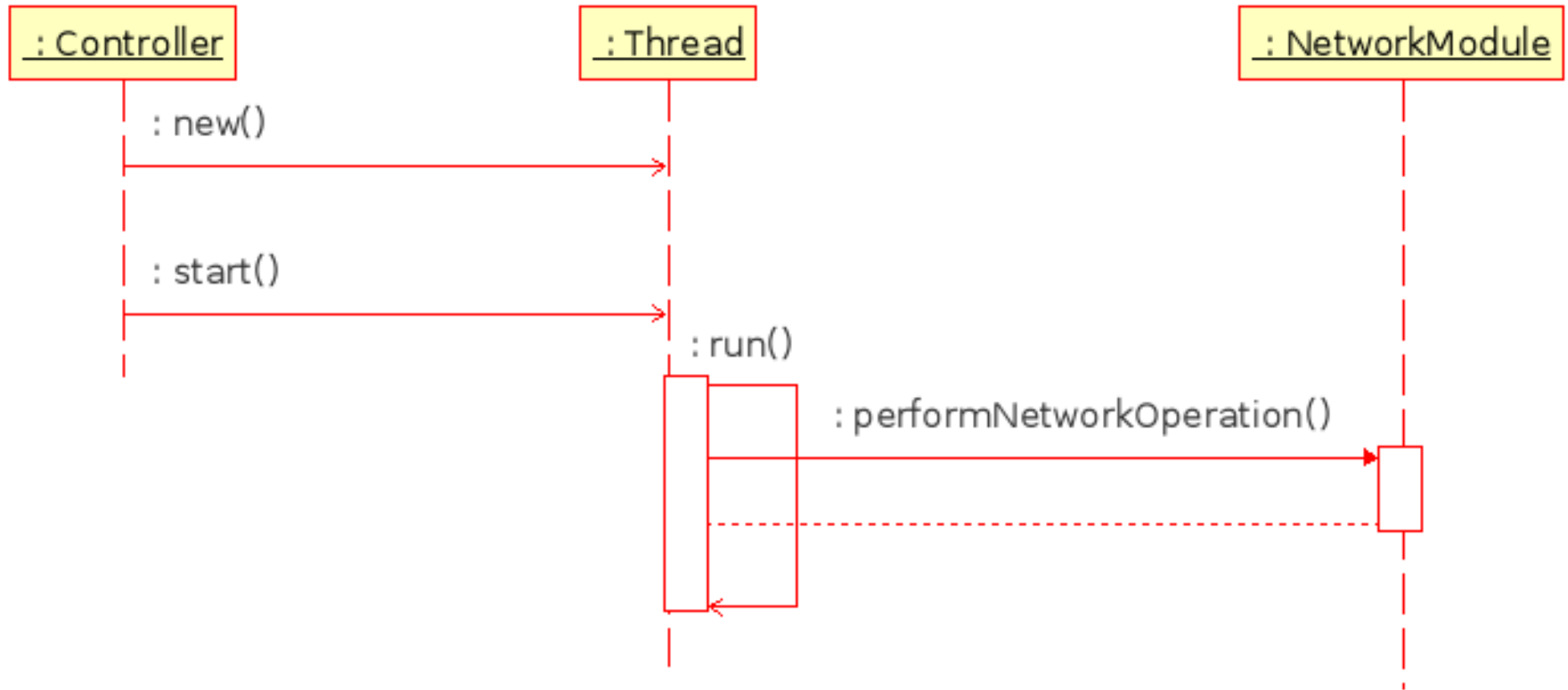
Il codice completo delle applicazioni d'esempio qui mostrate sarà pubblicato sul sito <http://www.gerdavax.it> entro il mese di ottobre 2006

- Malgrado la potenza delle ricche API opzionali, Java ME è carente nella gestione di due funzionalità essenziali per il MDP:
  - Framework multithreading task-oriented
  - Supporto per connessioni HTTP asincrone
  - Basic Authentication e POST-not-chunked
- Occorre predisporre opportune classi utility per ottimizzare le risorse (limitare il numero di thread), migliorare l'usabilità dell'applicazione, mantenere controllo completo dell'interfaccia utente e del sottosistema di comunicazione

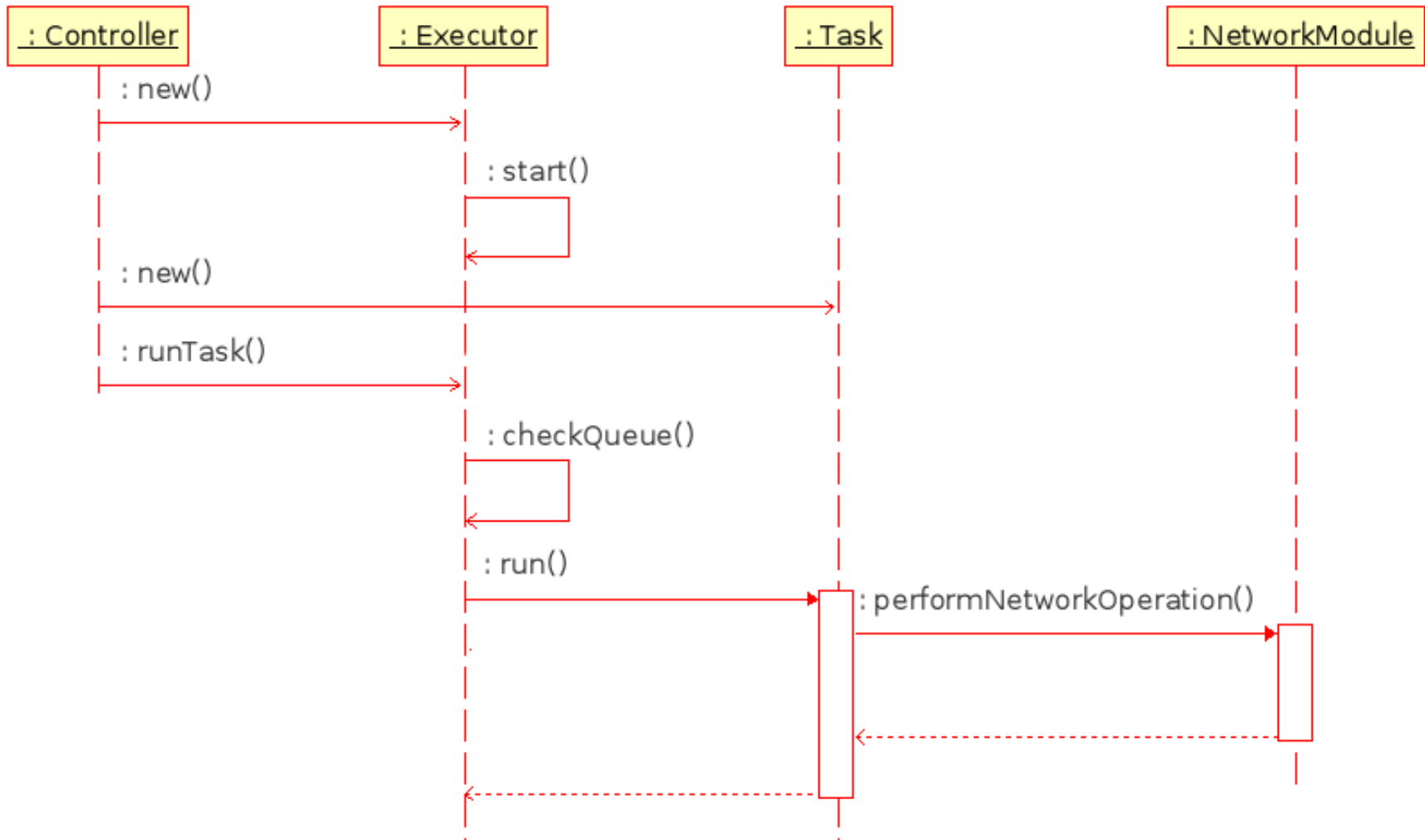
- Le operazioni di comunicazione (locale e remota) sono time-consuming e **bloccanti**
- Esempi:
  - **Networking**: startup GPRS, DNS lookup, risposta del server, trasferimento dati...
  - **Bluetooth**: discovery, pairing, transfer
  - **SMS**: incapsulamento e inoltro messaggio
- Le specifiche Java ME **raccomandano** l'uso thread per gestire tutte le operazioni di I/O in modo da separarle dal thread principale dell'applicazione e della interfaccia grafica

- **One-shot:** l'applicazione crea un thread “al volo” per svolgere un compito richiesto e prosegue nel resto dell'esecuzione
- **Single task:** l'applicazione dispone di un thread sempre attivo, al quale sono assegnati in sequenza i diversi task da elaborare. Se un task è attivo non può essere assegnato un nuovo task
- **Task queue:** l'applicazione dispone di uno o più thread sempre attivi e una coda dove i task da eseguire sono salvati in attesa che un thread sia libero per completarne l'esecuzione

# Thread... al volo!



# Multithreading task-oriented



- **Bloglines:**
  - servizio web che permette di aggregare e classificare i feed di interesse
  - <http://www.bloglines.com>
- Offre un servizio che permette di ricevere notifica in automatico in caso di aggiornamento dei feed di maggiore interesse
- Scenario mobile:
  - l'utente, connesso attraverso terminale mobile dotato di browser, ha necessità di ricevere notifica via RSS

- Java ME fornisce tutti gli elementi per implementare rapidamente il notifier per i servizi di Bloglines:
  - connettività http per l'accesso al servizio di notifica
  - multithreading per il controllo periodico del numero di articoli non letti
  - possibilità di avviare il browser locale (MIDlet.platformRequest()) per l'accesso ai servizi di lettura dei feed
    - Blogline fornisce una apposita interfaccia ottimizzata: <http://www.bloglines.com/mobile>

- Specifica: <http://www.bloglines.com/services/>
- La Notifier API è semplicissima ed efficace:
  - funzione:
    - <http://rpc.bloglines.com/update>
    - riceve come parametro l'account name (`user=ACCOUNT_NAME`)
    - restituisce il numero di articoli non letti tra i feed per i quali è attiva la notifica via Notifier nel formato `|NUMERO_ARTICOLI|`
  - esempio:
    - <http://rpc.bloglines.com/update?user=gerdavax@tiscali.it&ver=1> restituisce `|29|`

- BloglinesConnection:
  - gestisce la connessione con il servizio, effettua la richiesta e decodifica la risposta
  - notifica al listener quando ogni operazione di aggiornamento è completata o eventuali errori
- BloglinesConnectionListener:
  - riceve notifica delle operazioni eseguite dalla BloglinesConnection

# ConnectionListener

```
public interface ConnectionListener {  
  
    public void updated(int items);  
  
    public void failed(Exception e, String message);  
  
}
```



La MIDlet è un ConnectionListener: il risultato è asincrono rispetto alla richiesta e nessuna invocazione ai servizi di connettività è bloccante

# BloglinesConnection (1)

```
public class BloglinesConnection {
    private ConnectionThread connectionThread;
    private ConnectionListener listener;
    private String account;

    public BloglinesConnection() { init(); }

    private void init() {
        connectionThread = new ConnectionThread();
        connectionThread.start();
    }

    public void setAccount(String account) { this.account = account; }

    public void update() { connectionThread.connect(); }

}
```

# BloglinesConnection (2)


```
private class ConnectionThread extends Thread {
    private boolean halt;

    public void run() {
        // prossima slide...
    }

    public synchronized void connect() {
        notify();
    }

    public synchronized void halt() {
        halt = true;
        notify();
    }
}
```

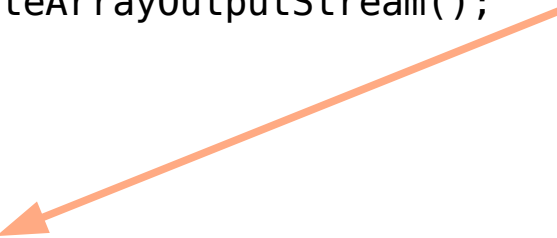
La richiesta di una nuova  
connessione sblocca il thread  
già creato in precedenza



# BloglinesConnection (3a)

```
public void run() {
    HttpURLConnection connection = null;
    InputStream input;
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    byte[] buffer = new byte[64];
    try {
        while (!halt) {
            synchronized (this) { wait(); }
            try {
                baos.reset();
                connection = (HttpURLConnection)
Connector.open(BloglinesProtocol.RPC_URL + "?user=" + account + "&ver=1");
                input = connection.openInputStream();
            }
            // SEQUE....
        }
    }
```

Appena avviato, il thread si blocca in attesa del comando di connessione (connect())



La HttpURLConnection permette di implementare immediatamente il protocollo

# BloglinesConnection (3b)

```
    if (connection.getResponseCode() == HttpURLConnection.HTTP_OK) {
        int read;
        while ((read = input.read(buffer)) != -1 && !halt) {
            baos.write(buffer, 0, read);
        }
        try { connection.close(); }
        catch (Exception e) { // nothing to do
        } finally { connection = null; }

        String result = new String(baos.toByteArray());
        int items = Integer.parseInt(result.substring(1,
result.indexOf("|", 1)));
        listener.updated(items);
    }
} catch (Exception e) { listener.failed(e, "Can't connect.");}
}
```

# In azione sul Nokia E61 (1)



# In azione sul Nokia E61 (2)



- Sacrificando la portabilità, è possibile sfruttare al massimo alcuni dispositivi
- I Nokia Series 60 e alcuni terminali di altri vendor permettono di spostare in secondo piano l'applicazione corrente, lasciandola in esecuzione e permettendole di tornare il primo piano quando necessario
  - Per andare in background:  
`display.setCurrent(null)`
  - Per tornare in foreground:  
`display.setCurrent(DISPLAYABLE_DA_MOSTRARE)`

- Come indicato sul sito, “Upcoming.org is a social events calendar driven by people just like you!”



- Squisitamente Web 2.0, permette di includere un widget con i **propri** eventi di interesse all'interno del proprio sito
- Offre una interfaccia REST per l'accesso ai servizi

# Il protocollo (1)

- E' descritto dettagliatamente nella sezione dedicata agli sviluppatori:
  - <http://upcoming.org/services/api/>
- L'applicazione necessita di una API Key per accedere ai servizi REST
- L'eventuale accesso a informazioni di uno specifico account è limitato dal possesso di un Token

- L'invocazione dei metodi avviene tramite http GET/POST:

```
http://www.upcoming.org/services/rest/?  
    api_key=<API_Key>&  
    method=event.search&  
    search_text=Symbian
```

- Le risposte sono documenti XML:

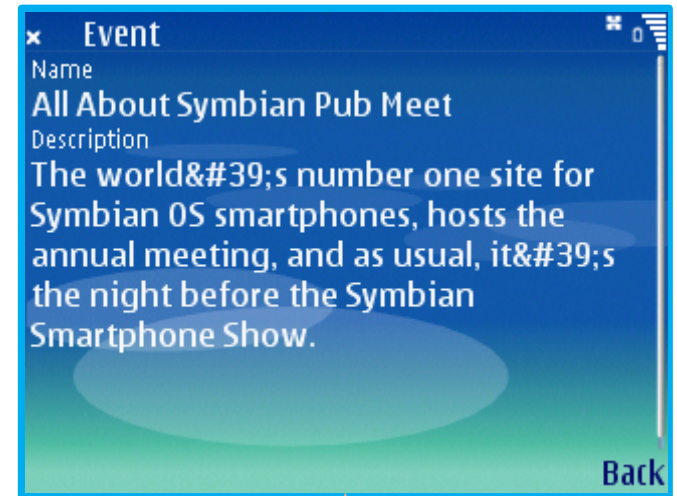
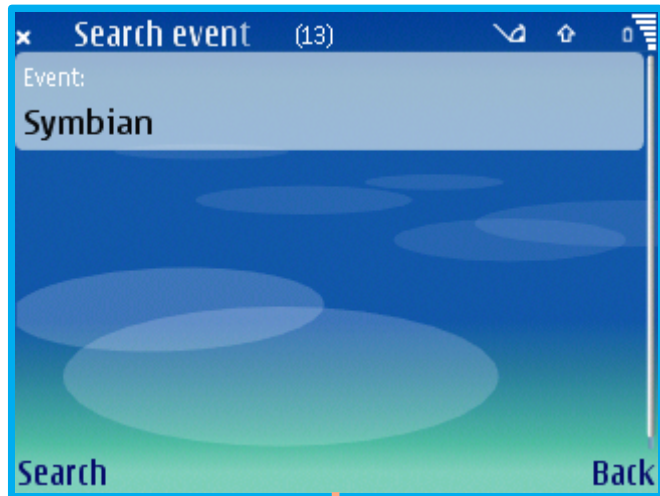
```
<rsp stat="ok" version="1.0">  
    [RISPOSTA XML, dipende dal metodo invocato]  
</rsp>
```

- Un esempio di risposta:

```
<rsp stat="ok" version="1.0">  
  <event id="87442" name="All About Symbian Pub Meet" description="The  
    world's number one site for Symbian OS smartphones, hosts  
    the annual meeting, and as usual, it's the night before the  
    Symbian Smartphone Show." start_date="2006-10-16" end_date=""  
    start_time="18:00:00" end_time="" personal="1" selfpromotion="1"  
    metro_id="49" venue_id="5417" user_id="62647"  
    category_id="4" date_posted="2006-06-25 11:37:36" latitude=""  
    longitude="" geocoding_precision="" geocoding_ambiguous=""/>  
  <event id="76389" name="The Smartphone Show" description="Organized  
    by Symbian" start_date="2006-10-17" end_date="2006-10-18"  
    start_time="" end_time="" personal="1" selfpromotion="0"  
    metro_id="49" venue_id="10928" user_id="55142" category_id="6"  
    date_posted="2006-05-08 12:22:55" latitude="" longitude=""  
    geocoding_precision="" geocoding_ambiguous=""/>  
</rsp>
```

- L'implementazione è molto simile a quella di Bloglines, con due integrazioni:
  - l'applicazione deve inviare il proprio API Key per accedere alle funzionalità e dovrà chiedere all'utente il Token al fine di accedere ai suoi dati personali
  - il risultato dell'elaborazione è un documento XML che occorre decodificare:
    - la JSR 172 è ormai presente in numerosi terminali Java ME e permette di scrivere un parser SAX 2.0
    - se non disponibile, si può usare la libreria kXml

# In azione su Nokia E61



- Sfruttando una connessione Bluetooth è possibile leggere i dati di posizionamento da un GPS ed effettuare **query georeferenziate**:
  - l'utente inserisce manualmente la keyword dell'evento di interesse
  - l'applicazione associa automaticamente i dati di posizione (latitudine/longitudine) alla query
- Java ME permette di implementare rapidamente una soluzione di questo tipo

- Attraverso la JSR 82 (Bluetooth API) è possibile aprire una connessione seriale:

```
gps = "btspp://0002C71536EA:1;authenticate=false;encrypt=false;master=false";  
StreamConnection conn = (StreamConnection) Connector.open(gps);  
buffer = new BufferedReader(new InputStreamReader(conn.openInputStream()));  
// leggiamo la riga:  
String nmea_sentence = buffer.readLine();
```

- Lo stream NMEA contiene le informazioni di interesse:

```
$GPGGA,164922.982,3859.4108,N,00856.1785,E,1,04,3.4,127.9,M,4  
7.6,M,0.0,0000*7A
```

↑  
Latitudine

↑  
Longitudine

↑  
Altitudine

←  
Satelliti

# La query georeferenziata

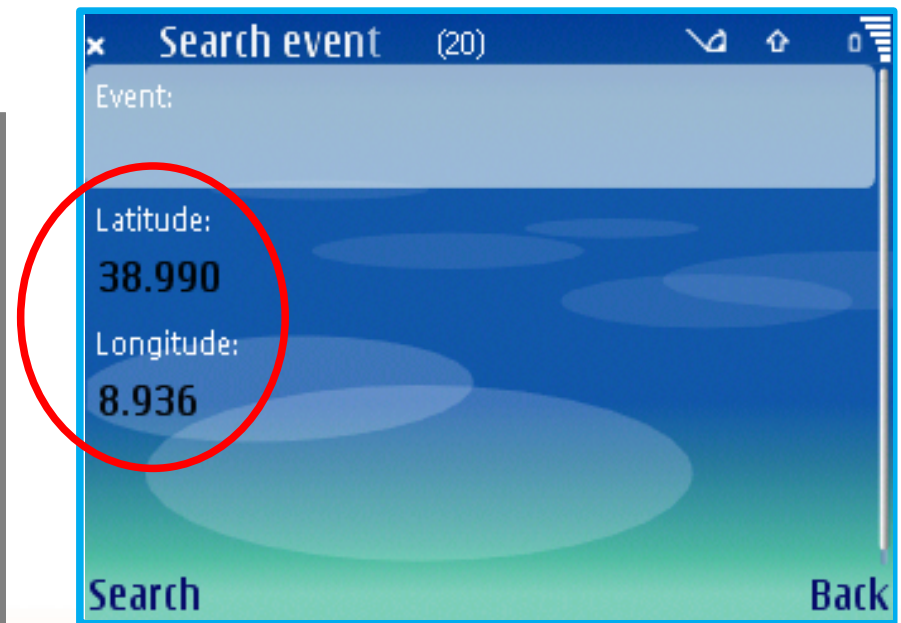
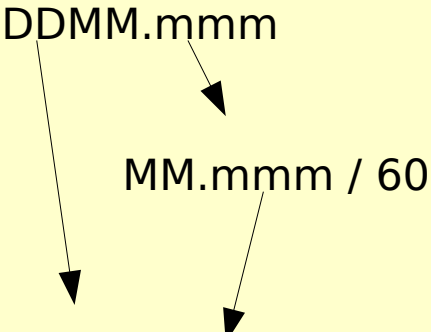
```
http://www.upcoming.org/services/rest/?  
api_key=API_KEY  
&method=event.search  
&query=Symbian  
&latitude=LAT&longitudeLON
```

Da NMEA a decimale:

NMEA: DDMM.mmm

MM.mmm / 60

Float: DD + .dddd



- Le applicazioni che fanno uso del Generic Connection Framework e che accedono a funzionalità quali PIM, Location e messaggista sono soggette alle restrizioni dell'Application Manager: l'utente deve **esplicitamente** consentire l'accesso a queste risorse alle MIDlet del dominio untrusted
- Per garantire accesso (quasi) incondizionato alle applicazioni è necessario firmarle digitalmente con un certificato valido e **riconosciuto dal terminale**

- Le applicazioni digitali firmate migliorano nettamente la user-experience sul dispositivo:
  - l'applicazione può accedere **autonomamente** alle risorse anche in background
  - l'utente non è costantemente messo di fronte al “bivio della fiducia” da assegnare alla MIDlet
  - alcune API sono accessibili solo ad applicazioni firmate (es: terminali Motorola)
- Dove acquistare un certificato digitale:
  - Verisign: <http://www.verisign.com>
  - Thawte: <http://www.thawte.com>

- Il Web 2.0 e la nuova piattaforma di sviluppo, che sposta il baricentro della rete:
  - verso gli utenti, **produttori dei dati**
  - verso i servizi componibili, e estendibili, **mezzi di fruizione dei dati**
- Java ME fornisce tutti gli elementi per integrarsi con la nuova piattaforma:
  - massimo supporto alla connettività locale e remota
  - ricca dotazione di API per multimedia, location, security, 2D & 3D...

- E' **la** community italiana degli sviluppatori Java ME
  - <http://jmdf.java2me.org>
  - <http://tech.groups.yahoo.com/group/jmdf>
- Nata nel 2005, conta già 140 iscritti!!!
- Attività:
  - mailing list per condivisione esperienze
  - organizzazione seminari tecnici
  - sviluppo progetti open source
- Prossimo evento:
  - Quarto Meeting JMDF, 16 Ottobre a Pisa



- Stefano Sanna
  - Website: <http://www.gerdavax.it>
  - Email: [gerdavax@tiscali.it](mailto:gerdavax@tiscali.it)
  
- Java Italian Association:
  - Website: <http://www.jia.it>

## Java ME e Web 2.0 nuovi scenari applicativi per dispositivi mobili

(Versione 1.1)

**(C) 2006 Stefano Sanna (gerdavax@tiscali.it)**

è garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation. Una copia della licenza in lingua italiana è disponibile presso:

**<http://www.softwarelibero.it/gnudoc/fdl.it.html>**

*•Tutti i marchi commerciali sono di proprietà dei rispettivi titolari e sono stati citati in questa presentazione a solo scopo illustrativo.*