

# Un sistema per il monitoraggio di impianti fotovoltaici

## Progetto e implementazione

Loris Fichera

Relatore: Prof. Corrado Santoro

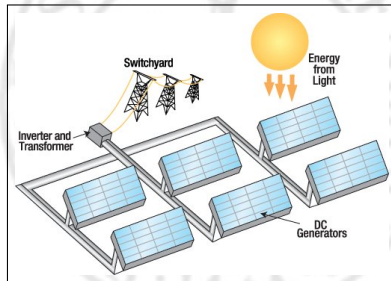
Università degli Studi di Catania  
Corso di Laurea Specialistica in Ingegneria Informatica

20 Luglio 2011

# Obiettivi

Un sistema di monitoraggio effettua la **raccolta** e **l'integrazione** dei **dati rilevanti** di un impianto al fine di determinarne:

- lo stato operativo
- l'efficienza globale
- la produzione energetica



Quali sono i **dati rilevanti**?

# Classi di utenti (Kolodenny 2008)

Kolodenny *et al.*, identificano le seguenti **classi di utenti**:

- **ricercatore**

- quanti più dati possibile
- alto livello di dettaglio

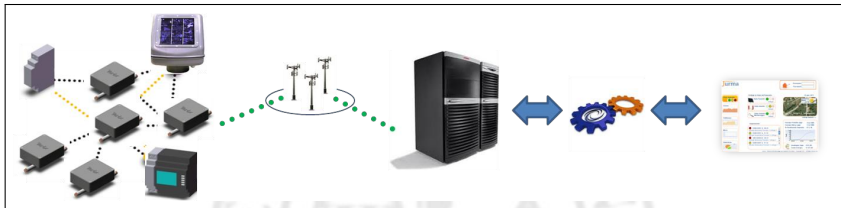
- **proprietario**

- energia prodotta
- ritorno economico

- **manutentore**

- comportamento di ogni componente (modulo, inverter...)
- dati ambientali

## Panoramica del sistema



- infrastruttura di campo
- trasmissione dati via gprs (o sms)
- datacenter
- accesso ai dati via web services
- portale web

- └ La soluzione implementata
- └ L'infrastruttura di campo

## Infrastruttura di campo

- **Wireless Sensor Network (WSN) ZigBee-based**
  - bassi costi
  - facilità di installazione
    - no cablaggi aggiuntivi
    - rete autoconfigurante
- previsti tre tipi di nodi
  - gateway
  - power/inverter transponder
  - string transponder

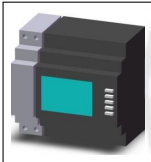
- └ La soluzione implementata
- └ L'infrastruttura di campo

# Nodi WSN



## ■ Gateway

- concentratore dati
- trasmette dati via **gprs/sms**
- effettua misure di irraggiamento



## ■ Power/Inverter Transponder

- misura di **grandezze elettriche** (corrente, tensione, PF...)



## ■ String Transponder

- misura della **corrente di stringa**
- sensore a **effetto Hall**

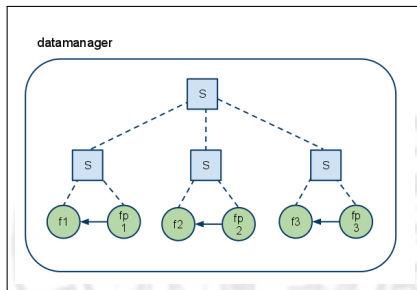
## Il Datacenter

Si occupa di

- gestire i flussi informativi generati dai gateway
  - elaborare e memorizzare i dati di campo
  - generare report e allarmi
  - fornire accesso ai dati mediante web services RESTful
- 
- è interamente basato su Erlang/OTP
  - è costituito da un insieme di Erlang applications



Gestisce i flussi informativi generati da ogni impianto



- un *process pool* per ogni impianto
- due **worker** per *pool*
  - un filter
  - un processo **endpoint** (`file_poller`, `sms_manager`)



## Il file\_poller

- decodifica i dati ricevuti via **FTP**
- genera **report** riguardo
  - dati **corrotti**
  - dati **mancanti**
- invia i dati ricevuti al filter

### Decodifica dei dati

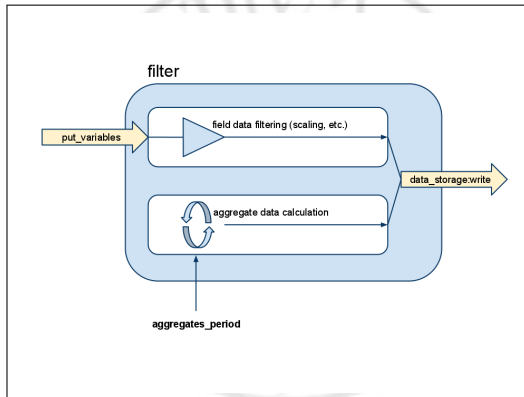
```
try
  {ok, FileContent} =
    file:read_file (lists:concat ([Path, File])),
    ftp_protocol:decode (FileContent)

catch error : {Reason, D} ->
  ...
  nil
end,
```

## Il filter/1

Due attività concorrenti:

- **filtraggio** dei dati ricevuti (scaling, ecc.)
- **stima** di altri valori

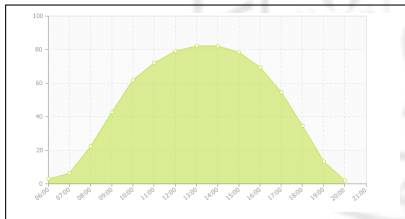
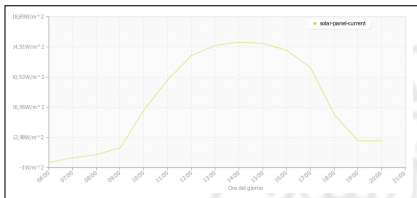


## Il filter/2

### Calcolo della potenza totale per Power Transponder

```
{ok, {_, Vars}} =  
    data_storage:get_trend_by_device (DeviceID, F, T,  
                                      ["active-power-l1",  
                                      "active-power-l2",  
                                      "active-power-l3"]),  
  
PropList = measure_data_to_proplist (Vars),  
P1 = proplists:get_value ("active-power-l1", PropList),  
P2 = proplists:get_value ("active-power-l2", PropList),  
P3 = proplists:get_value ("active-power-l3", PropList),  
  
TotalActivePower = P1 + P2 + P3,
```

# Radiazione vs. Potenza prodotta



## Configurazione

- 1 Power Transponder
- 1 Gateway

## Valori di picco (1400hrs)

- ~ 900 W/m²
- ~ 82 kW

## Sviluppi futuri

- **sicurezza** delle comunicazioni Gateway - Datacenter
- **sistema esperto** per la diagnosi delle anomalie

