



# **Powersim Software Development Kit (SDK)**

*Descrizione dell'architettura necessaria per gestire modelli Powersim server based*

*Maggio 2008*

# Obiettivo del documento

L'obiettivo del presente documento è di descrivere le caratteristiche della soluzione architeturale per gestire modelli di simulazione Powersim in ambiente server based.

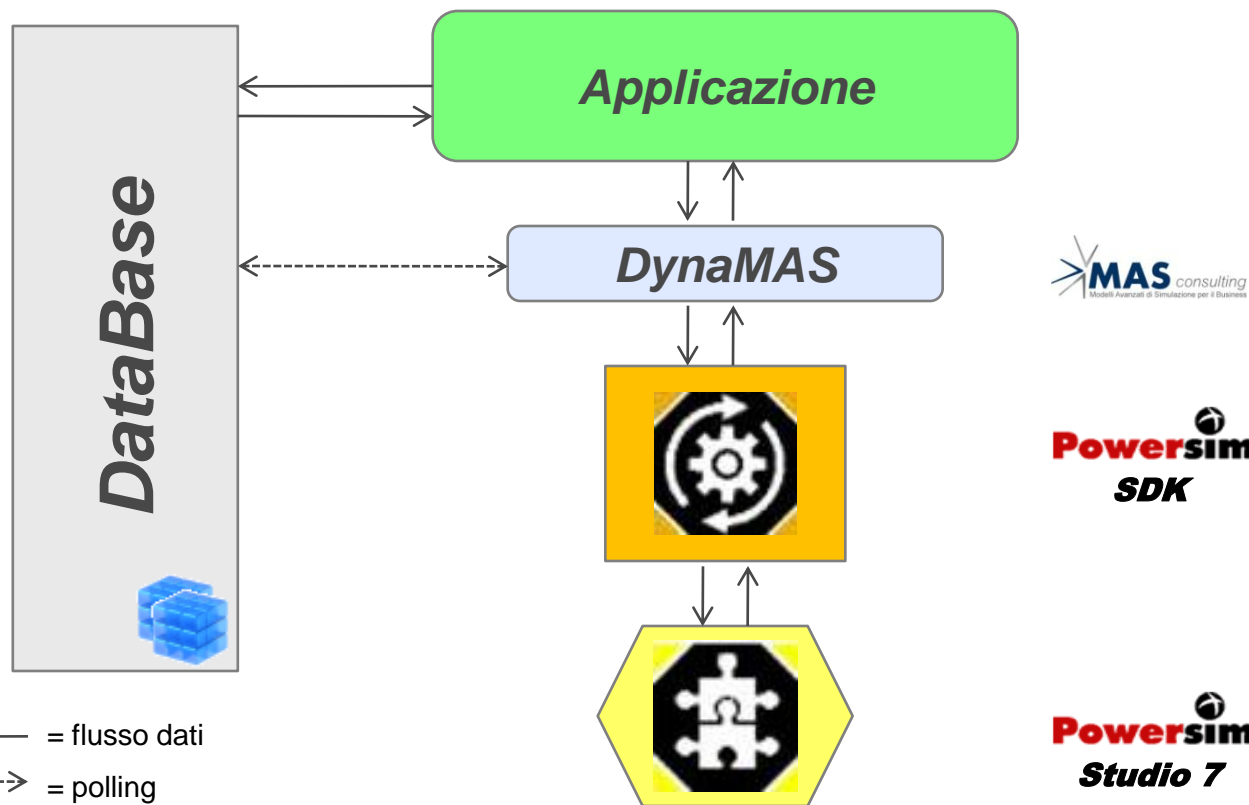
In particolare gli ambiti di analisi sono i seguenti:

- Architettura Applicativa;
- Powersim SDK: Overview;
- Powersim SDK: Principali finalità;
- Powersim SDK: Modalità di utilizzo.

# Architettura Applicativa

Powersim SDK permette di gestire i modelli di simulazione costruiti utilizzando Powersim Studio.

## Struttura applicativa della soluzione Powersim SDK



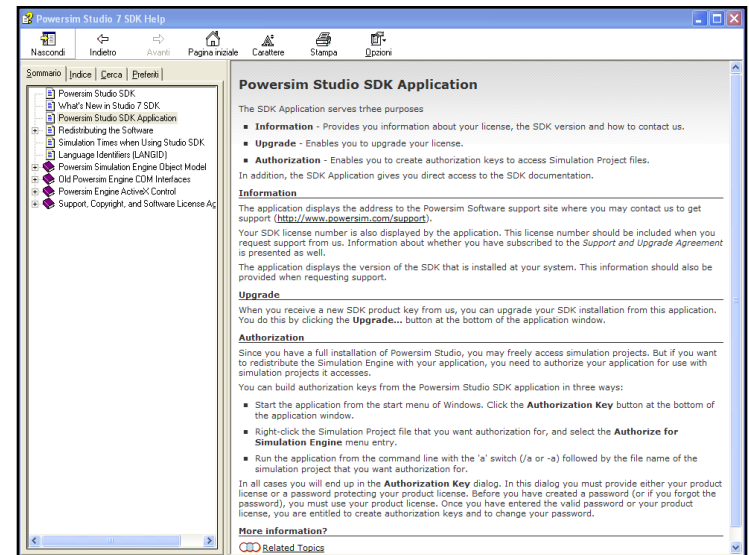
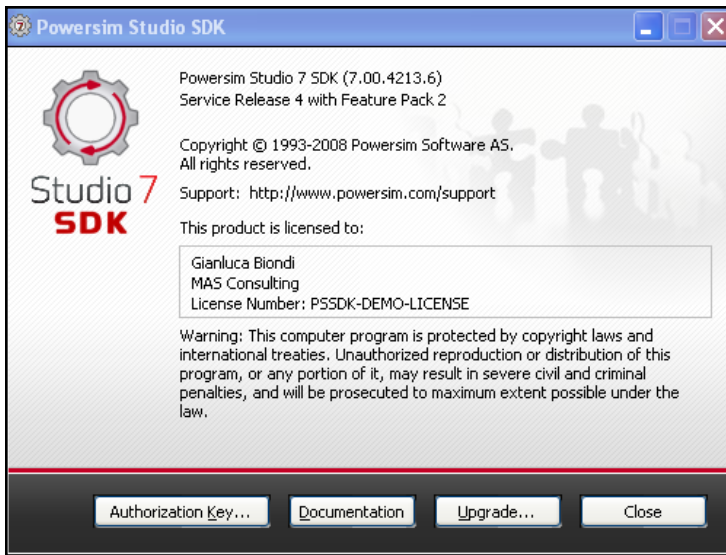
### Descrizione

- **Applicazione** : realizzata in qualsiasi modo (es. Visual Basic, pagine web, etc...). Lo sviluppo può essere effettuato dal personale del cliente con la collaborazione di MAS
- **DynaMAS**: tool che consente la gestione della simulazione utilizzando gli oggetti di Powersim SDK
- **SDK**: gestisce il motore di calcolo costruito mediante Powersim Studio
- **Powersim Studio**: ambiente in cui vengono costruiti i modelli di simulazione
- **Database**: contiene tutti i dati necessari alle simulazioni.

# Powersim SDK: Overview (1/2)

Powersim SDK è uno strato applicativo che permette l'interazione con un modello Powersim senza l'utilizzo del software Studio 7.

Powersim Studio 7 viene usato per creare e modificare il modello (file .sip), che viene poi utilizzato con Powersim SDK. SDK è un insieme di librerie che vengono installate lato server.



Powersim SDK riconosce e gestisce il modello di simulazione che viene istanziato sul server

```
set MyProject = PsEngine.OpenProject("C:/Powersim/Acquirente Unico/Prezzo_cessione.sip")
```

## Powersim SDK: Overview (2/2)

Powersim SDK interagisce con il modello di simulazione attraverso il *Simulation Engine Object Model*, che replica la totalità delle dinamiche e degli attributi costruiti con Studio 7, mantenendone la struttura a oggetti.

Il *Simulation Engine Object Model* sostituisce e amplia le librerie COM e gli oggetti ActiveX utilizzati dalle precedenti versioni di Powersim SDK.

### Perché usare SDK?

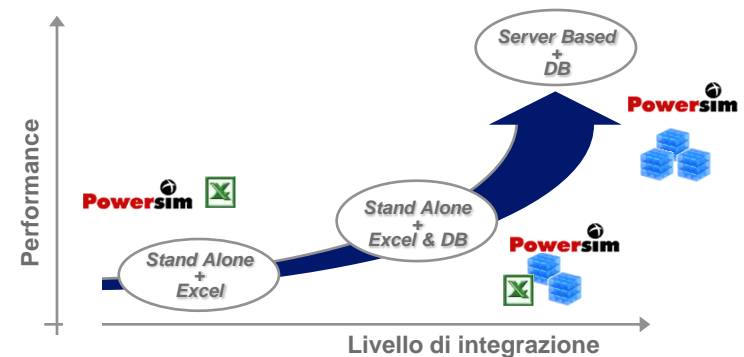
- E' lo strumento indispensabile per garantire l'integrazione, con gli altri sistemi aziendali, dei modelli realizzati con Powersim Studio;
- Permette il raggiungimento di performance molto più elevate rispetto alle installazioni Client, soprattutto per calcoli onerosi (Risk Analysis, Optimization Process);
- Garantisce una completa distribuzione del modello sviluppato. Gli utenti non necessitano di Powersim per utilizzare il modello, bensì possono utilizzare qualsivoglia tipologia di applicazione (es. interfaccia Visual Basic, etc.). Vantaggio: futuri risparmi sulle licenze;
- Garantisce un repository di dati centralizzato e controllato ed un continuo allineamento tra tutti gli utenti.

# Powersim SDK: Principali finalità

Tramite la migrazione a Powersim SDK è possibile:

- Gestire centralmente modelli Powersim e distribuirli agli utenti collegati in rete;
- Connettere nativamente il flusso dati (input e output) tra modelli sviluppati in Powersim Studio e DB esistenti;
- Gestire la simulazione in duplice modalità:
  - Simulazione “one-shot”: selezione di uno specifico scenario (set di dati in input + leve selezionate) e run della specifica simulazione;
  - Risk Assessment: creazione di n scenari, legati a specifiche distribuzioni di probabilità, e esecuzione di n simulazioni relative (es. metodo MonteCarlo);
- Raggiungere performance adeguate alle esigenze di business attuali e future..

Powersim SDK consente di far evolvere i modelli sviluppati in Studio da un ambiente totalmente stand-alone ad un ambiente server-based.



# Powersim SDK: Modalità di utilizzo (1/4)

Powersim SDK deve essere “programmato”.

Gli oggetti e i metodi nativi di Powersim SDK necessitano di un ulteriore strato applicativo (Service Layer - DynaMAS) che permetta l’interazione tra l’ambiente esterno e le librerie SDK.

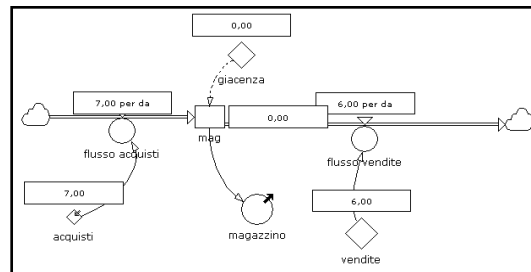
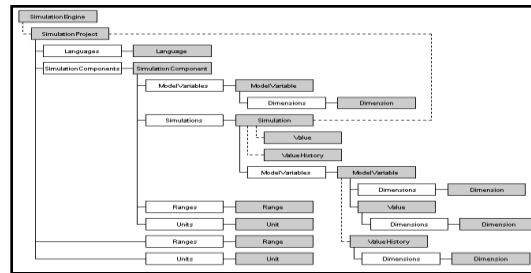
La tecnologia di realizzazione del Service Layer dipende dall’infrastruttura informatica attualmente a disposizione.

MAS Consulting vanta pluriennale esperienza nel campo dell’utilizzo di Powersim SDK.

Acquisti	<input type="text" value="7.0"/>
Vendite	<input type="text" value="6.0"/>
Giacenza	<input type="text" value="0.0"/>
Simulation Status:	End of simulation, click <a href="#">Reset</a> to restart model!
Simulation Time:	722520
<input type="button" value="Run intera simulazione"/>	
<input type="button" value="Run Step by Step"/>	Numero di step per volta: <input type="text" value="1"/>
<input type="button" value="Reset"/>	

```
<? Language=VBScript %>
<!--#include file="include/WebengineFunctions.asp"-->
<!--#include file="include/OutputFunctions.asp"-->
<!--#include file="include/SettingsFunctions.asp"-->
<!--#include file="include/Util.asp"-->

<X
dim PSEngine
set PSEngine = CreateObject("PsSimEng.SimulationEngine.1")
dim MyProject
set MyProject = PSEngine.OpenProject("C:/Powersim/Acquirente Unico/Prezzo_cessione.1")
dim MySim, MyVar, MyVal
set MySim = MyProject.OpenDefaultSimulation
set MyVar = MySim.Variables
%>
```



## Presentation Layer



## Service Layer - DynaMAS



## Powersim SDK Layer

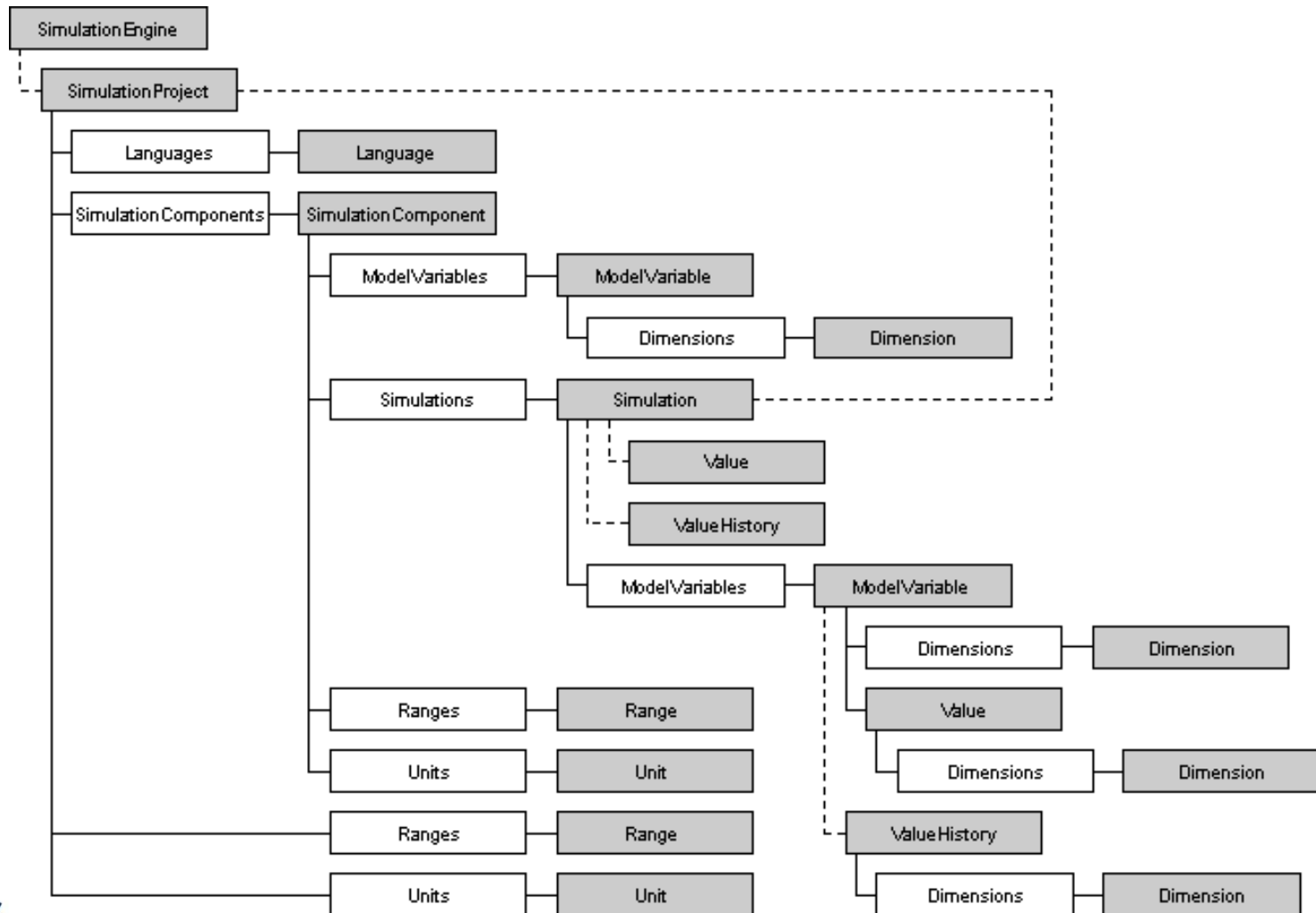


## Simulation Engine



# Powersim SDK: Modalità di utilizzo (2/4)

## Simulation Engine Object Model.



# Powersim SDK: Modalità di utilizzo (3/4)

Utilizzando il Simulation Engine Object Model è possibile:

- Aprire un progetto di simulazione e istanziare tutti gli oggetti da usare;
- Leggere e impostare la dinamica temporale: start time, stop time e time step;
- Impostare e/o estrarre i valori per ogni variabile appartenente al modello;
- Lanciare, arrestare, resettare la simulazione;
- Lanciare la simulazione per singoli step e seguire l'andamento delle variabili;
- Estrarre le serie temporali per le variabili time-dependent;
- Leggere l'attuale momento della simulazione (e lanciare un'azione come conseguenza);
- Ricavare lo stato della simulazione e attivare controlli (log, messaggi di errore, esito, etc.);
- Impostare la lunghezza della simulazione (non sempre serve attivare l'intera simulazione);
- ...e molto altro ancora!!!

Da...



...a...

```
for i = 1 to intNumSteps  
doEachStep  
Status = MySim.advance  
if Status = 0 Then
```

...mantenendo  
ogni funzionalità

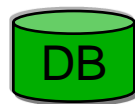
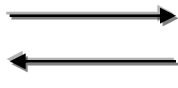
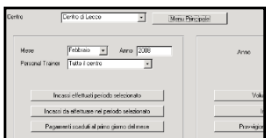
# Powersim SDK: Modalità di utilizzo (4/4)

## Il Simulation Engine Object Model:

- Può essere utilizzato in diversi ambienti di sviluppo: Visual Basic .NET, Visual C++, Visual C#, ASP.NET e VB Scripts;
- Può interagire con qualsiasi applicazione che supporti VBA (Microsoft Office);
- Può essere richiamato in pagine HTML o ASP.

Vengono spesso utilizzate combinazioni di più ambienti, ad esempio:

Applicazione VB che imposta le variabili e “ordina” il lancio della simulazione



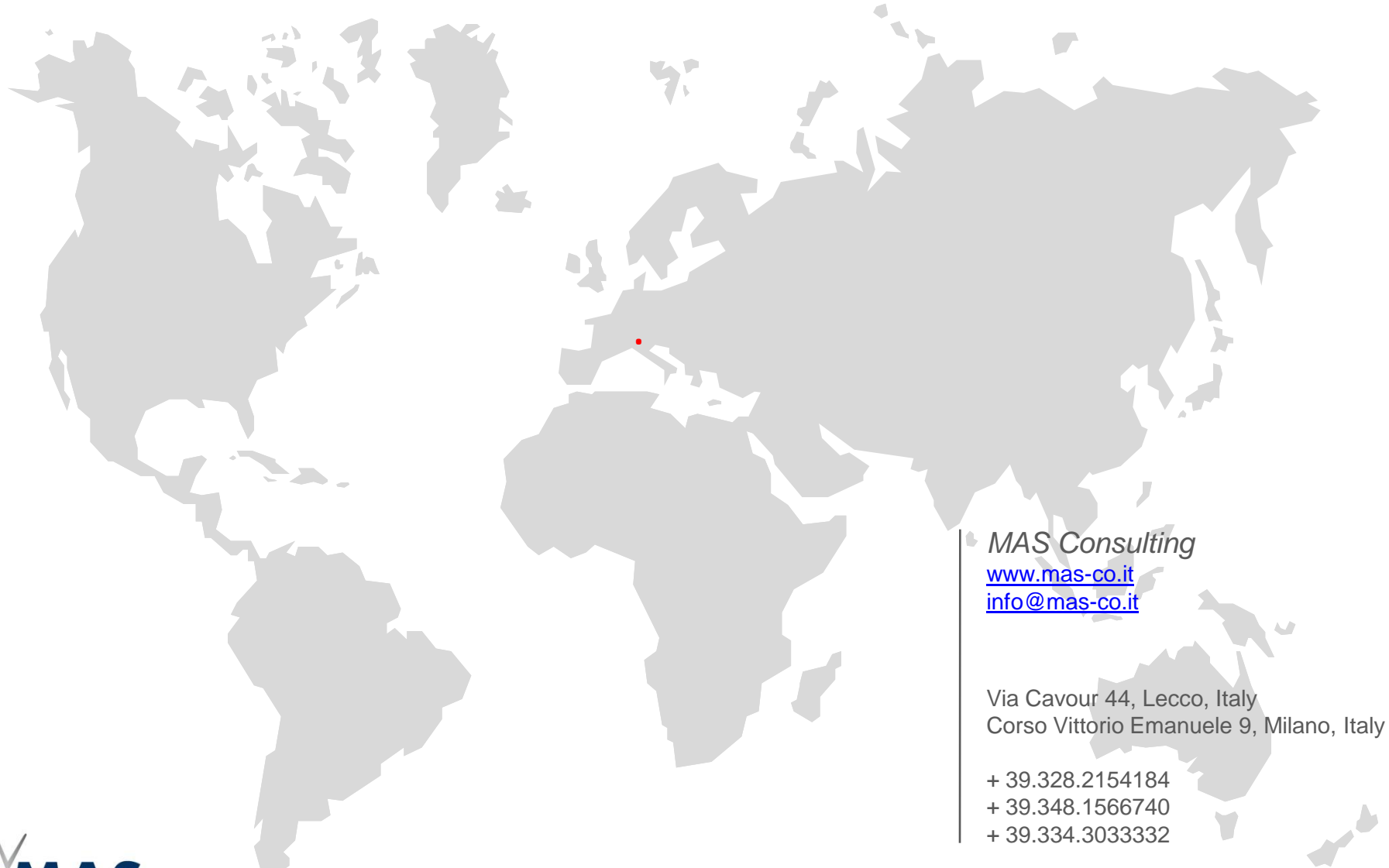
Servizio Windows di polling su DB e lancio simulazione tramite Powersim SDK



Motore di simulazione Powersim



# Contatti



*MAS Consulting*

[www.mas-co.it](http://www.mas-co.it)

[info@mas-co.it](mailto:info@mas-co.it)

Via Cavour 44, Lecco, Italy  
Corso Vittorio Emanuele 9, Milano, Italy

+ 39.328.2154184

+ 39.348.1566740

+ 39.334.3033332