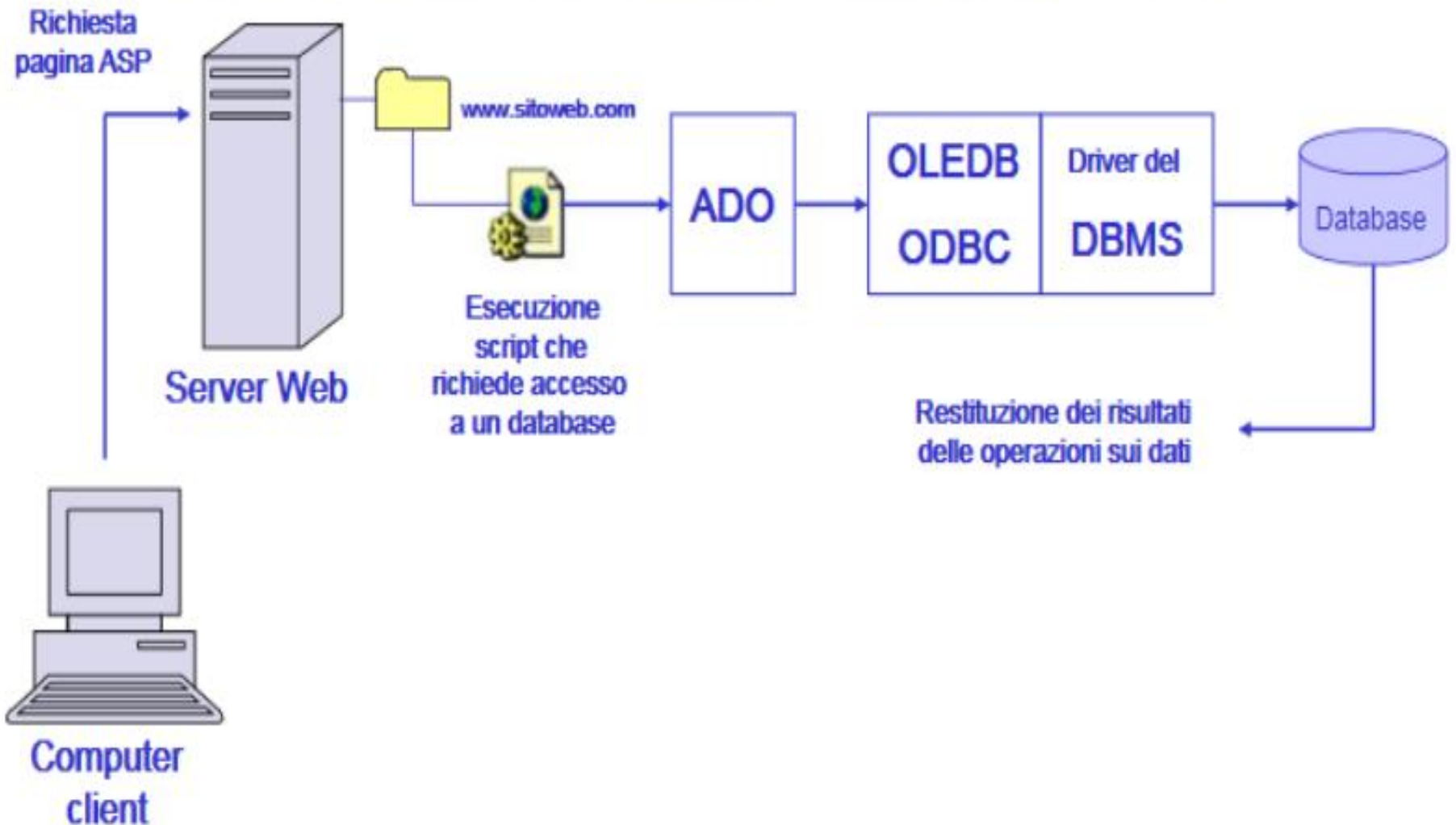
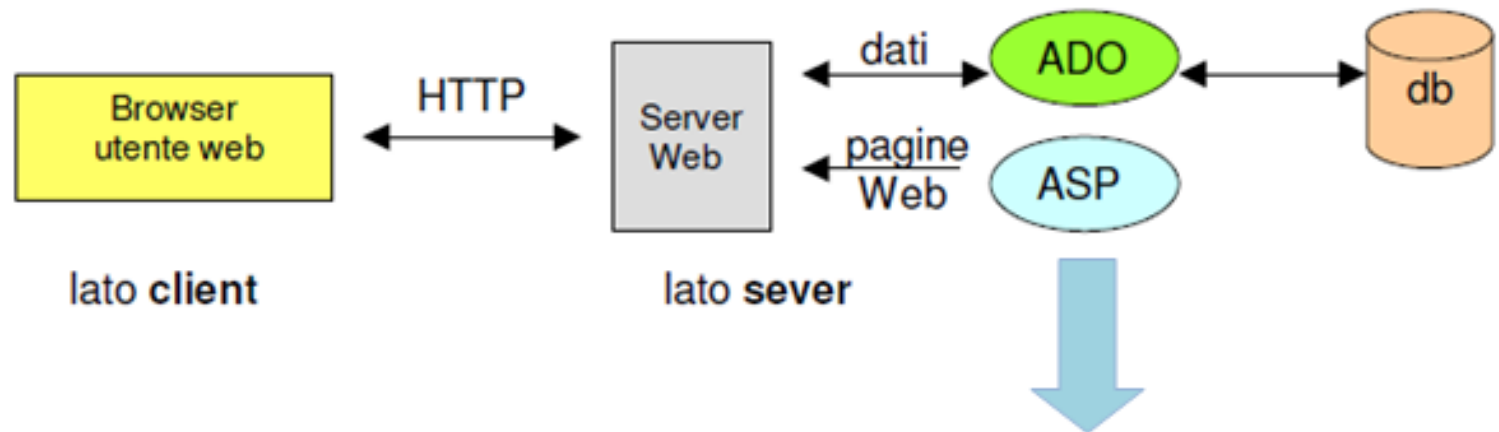
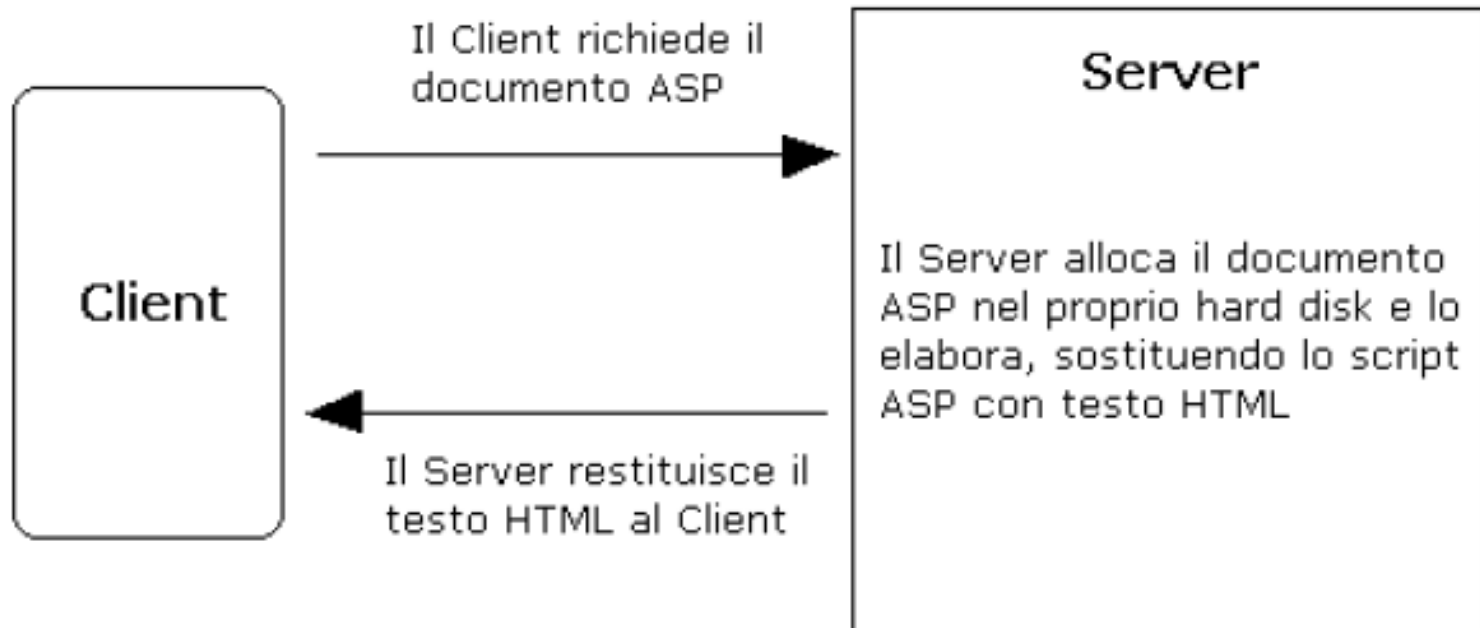


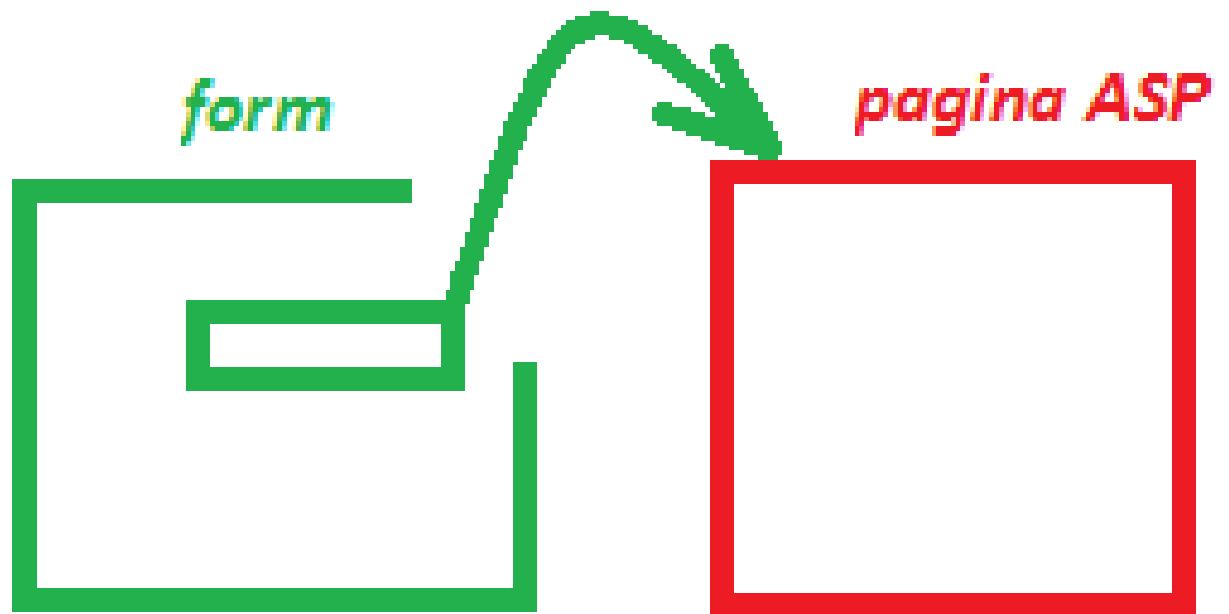
Schema delle Interfacce software per l'accesso a un database mediante una pagina ASP



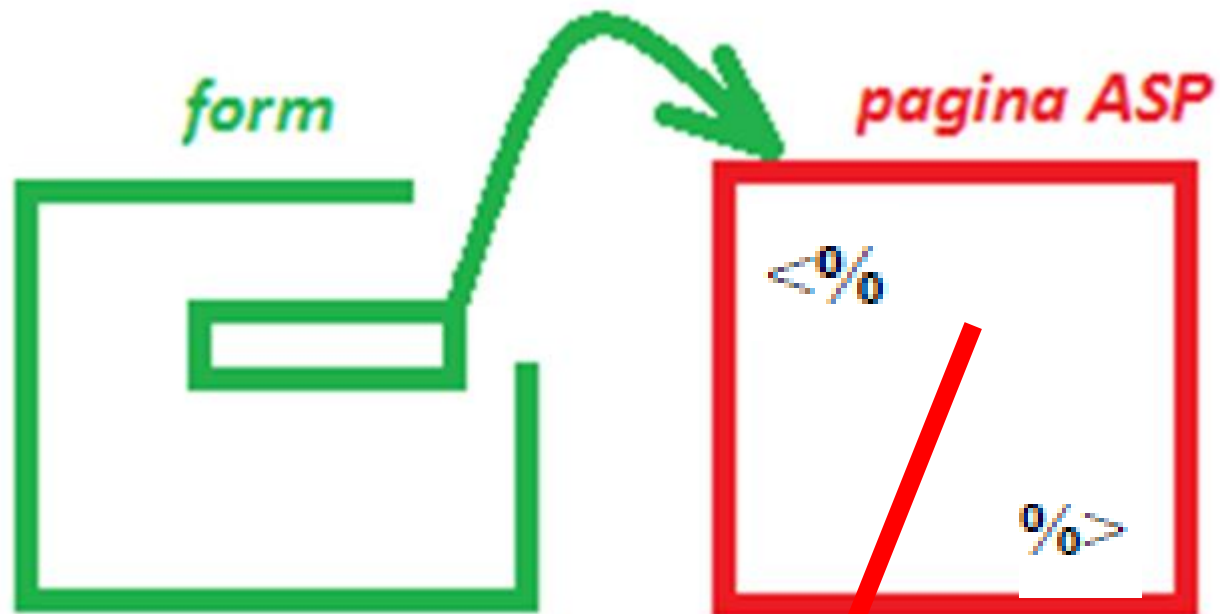


Interazione Client/Server per documenti ASP





```
<input type="text" name = "NAMEvalore">
```



name = "NAMEvalore"

per estrarre il valore in risposta ad un metodo **GET**:

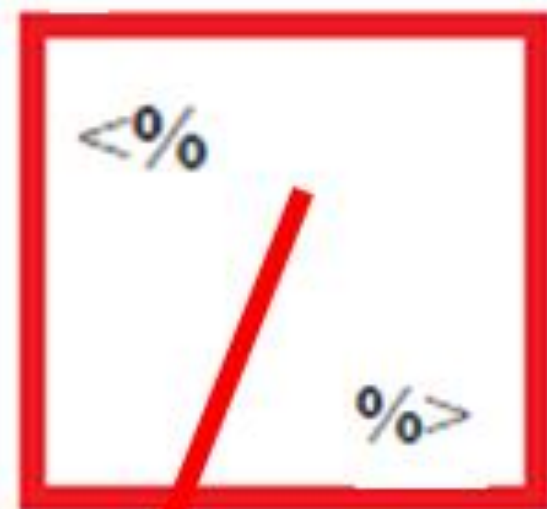
```
REQUEST.QUERYSTRING ("NAMEvalore")
```

per estrarre il valore in risposta ad un metodo **POST**:

```
REQUEST.FORM ("NAMEvalore")
```

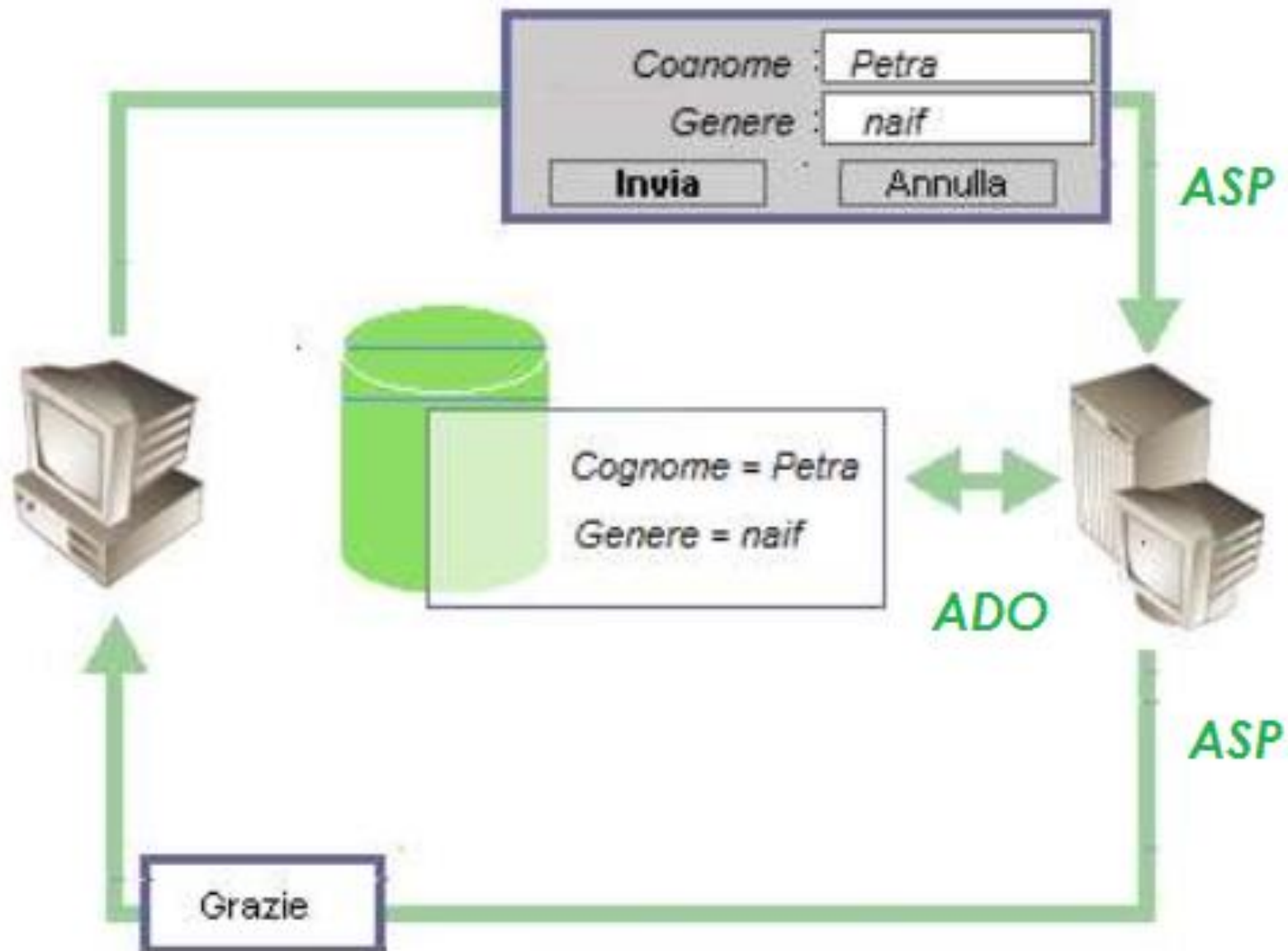


pagina ASP

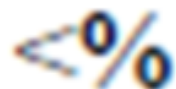


Grazie

`RESPONSE.WRITE ("Grazie")`



pagina ASP

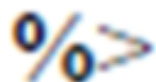


- ' creare un oggetto di tipo connessione
- ' creare un oggetto di tipo recorset
- ' aprire la connessione impostando la stringa (DSN-less)

- ' salvare come stringa il comando SQL
- ' eseguire la query a connessione aperta

- ' inviare al browser risultato

- ' chiudere connessione e deallocare oggetti ADO



ADO

Per gestire i database con tecnologia **ASP** si utilizzano **strumenti ADO** (ActiveX Data Objects): un'architettura che fornisce oggetti di **alto livello** per l'**accesso universale ai dati** e in particolare contiene **oggetti per la comunicazione con i database**.

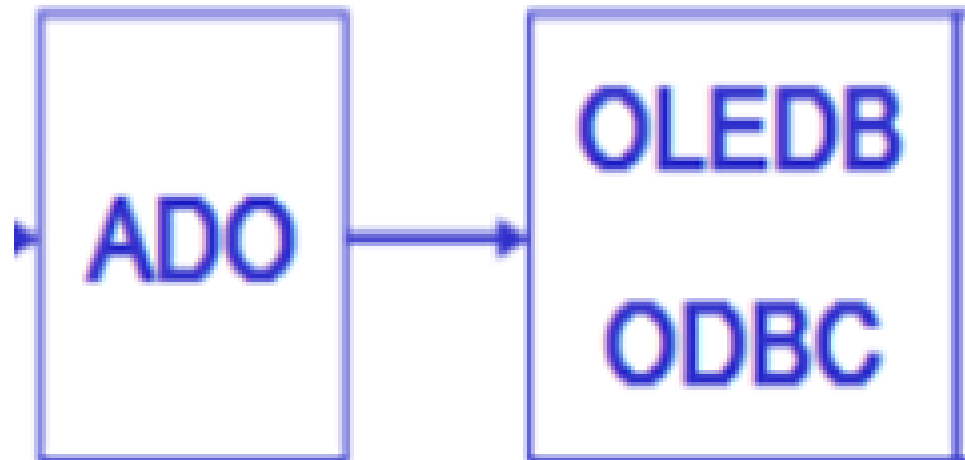
ADO: ideato come interfaccia dati C/S, consente di scrivere un'applicazione per la gestione e l'accesso ai dati contenuti in un server di database tramite un provider **OLE DB**

OLE DB

Interfaccia a **basso livello** che fornisce un modello di accesso ai dati universale che ne consente la gestione indipendentemente dal formato e dal metodo di memorizzazione; permette l'accesso non solo ai database relazionali ma a qualsiasi fonte dati : database **locali** o **remoti**, non relazionali, sistemi di file, posta elettronica, testo, grafica e oggetti multimediali, aziendali, personalizzati ...

Limiti di ADO

Una limitazione di ADO: impossibilità di creare fonti di dati **ODBC** pur se il provider predefinito è Microsoft OLE DB per ODBC (componente Microsoft Windows ed in particolare di Windows Open Services Architecture, che consente l'accesso a tutti i tipi di database relazionali).



Stringa di connessione

Poiché tuttavia ciascun provider dispone di caratteristiche uniche, le modalità di interazione tra l'applicazione e ADO variano leggermente a seconda del provider stesso (in particolare, è specifica la *stringa di connessione*).



Per una descrizione più completa dell'architettura **ADO** si consulti: **Il Modello ad Oggetti di ADO** (documento compresso scaricabile **Download il file Zip**)

- Tutorial (in inglese)
<https://www.w3schools.com/default.asp>
- ASP – HOME
<https://www.w3schools.com/asp/default.asp>
– Classic ASP / ASP.NET Web Forms / ASP.NET MVC /
ASP.NET Web Pages / ASP.NET API / ASP.NET Core
- ASP Classic
https://www.w3schools.com/asp/asp_introduction.asp
- ADO https://www.w3schools.com/asp/ado_intro.asp

Framework ASP.NET



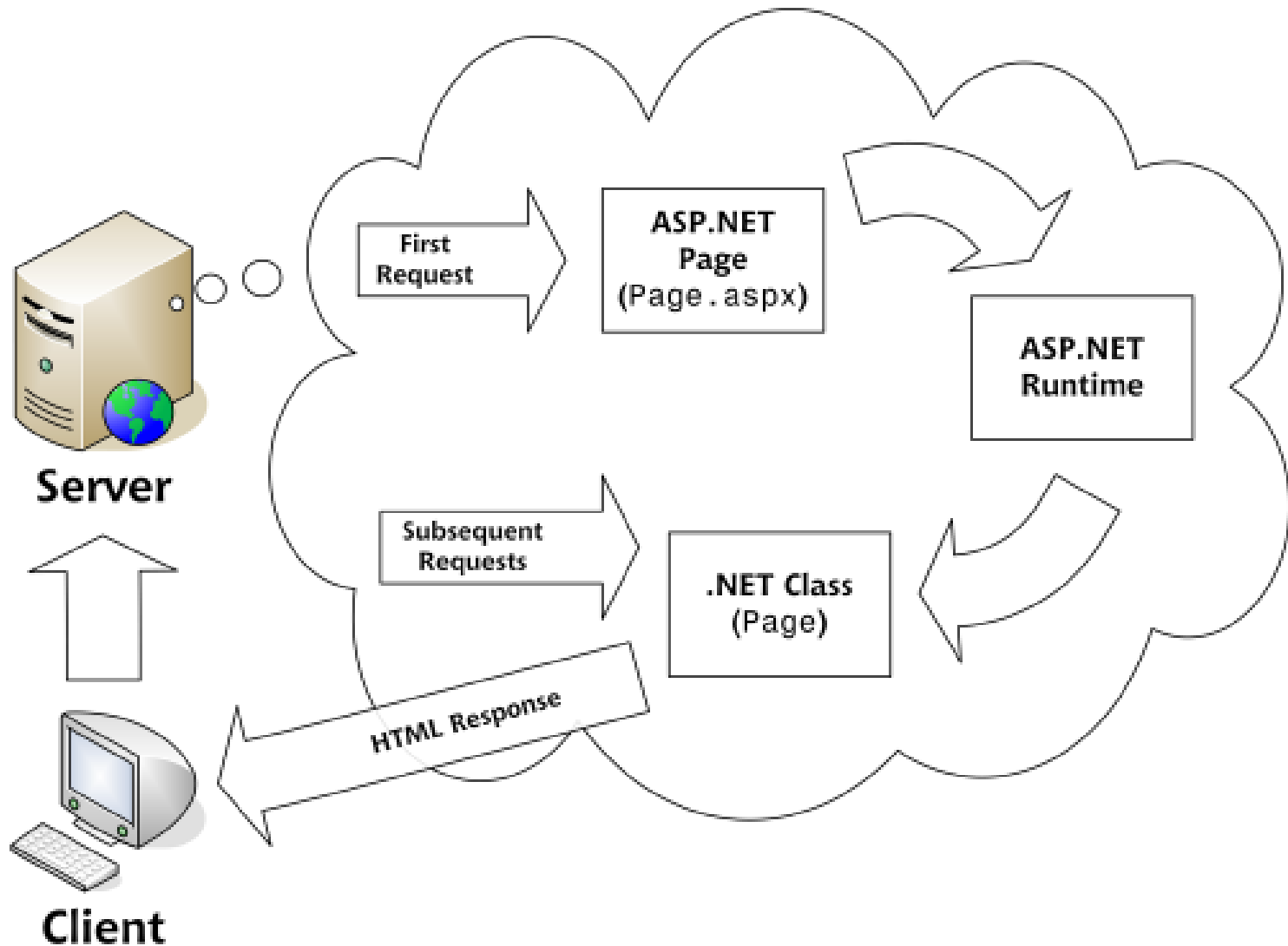
ASP.NET è un insieme di tecnologie di sviluppo di software per il web, commercializzate da Microsoft.

Framework ASP.NET

*Uno dei prodotti della suite che permette agli sviluppatori di realizzare applicazioni Web e servizi Web (Web Service) utilizzando uno qualsiasi dei **linguaggi di alto livello** supportati dal **Framework .NET** (libreria di classi) , come, ad esempio, Visual Basic.NET (altro prodotto della suite - proprietario Microsoft), **C#**, **J#** (derivato da Java), ma anche, molti altri linguaggi **open source** (e non), come Perl e Python.*



Page ASPX



Asp vs Asp.NET

Con la nascita del framework .NET, la tecnologia ASP non riceve più aggiornamenti e spesso si specifica come **ASP classica** nel confronto con la nuova suite che, pur con lo stesso nome, introduce sostanziali differenze.

- **Process Isolation**

- Pagina ASP stesso processo IIS (Microsoft Internet Information Services)
- Pagina ASPX: processi indipendenti

- **Interpretation vs. Compilation**

- Pagina ASP interpretata al volo
- Pagine ASPX sempre compilate in classi NET

*ASP.NET permette l'uso di un qualsiasi linguaggio ad alto livello della suite.
(code-behind)*

References

[wikipedia:Active Server Pages](#)
[wikipedia:ASP.NET](#)

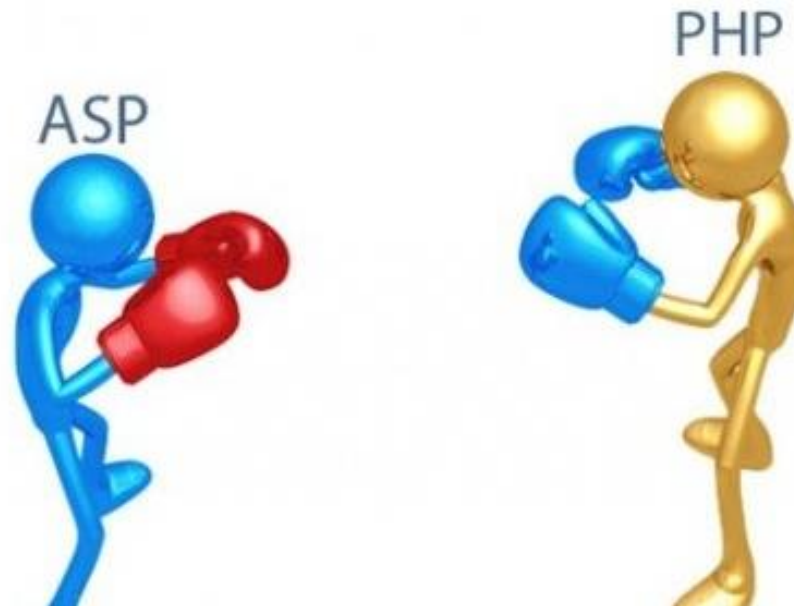
ASP *classico*: caratteristiche

- Le pagine sono completamente integrate nei file HTML.
- Non necessitano di compilazione: il motore o interprete è una **dll** caricata dal Web-Server
- Sono orientate agli oggetti ed usano componenti server ActiveX.
- Può essere utilizzato un qualunque *linguaggio di scripting* del quale sia installato sul web server lo *scripting engine*.

La tecnologia ASP supporta in modo nativo VBScript e Jscript

- Un'*applicazione* non è altro che una **directory** interna alla home page per la quale è attivo il **permesso di esecuzione**.
- È possibile modificare il linguaggio predefinito sia a livello di Web Server, cioè per tutte le applicazioni ASP gestite da IIS, che a livello di singola applicazione, cioè per tutte le pagine che la compongono

Perché ASP.NET non è la scelta migliore per realizzare una applicazione Web?



Articolo (30 ottobre 2014)

<https://www.gardainformatica.it/blog/sviluppo-software/perche-asp-net-non-scelta-migliore-per-realizzare-applicazione-web>

PHP vs ASP.NET

"Quale tecnologia è la migliore *Asp.net* o *PHP*?"

.... in post e articoli recenti, che confrontano **tecnologie server side**, riferimenti ad ASP sono spesso da intendersi con riferimento al framework Asp.NET (rilasciato nel 2002 come successore del **Classic ASP**)

Articolo (7 settembre 2013 – link [inattivo](#))

- [Per confronto : migrare da ASP a PHP](#)
- [ASP.NET vs PHP \(in inglese\)](#)



Unico pregio ?

- Le applicazioni ASP.NET sono significativamente più veloci e performanti rispetto a quelle realizzate utilizzando altre tecnologie di scripting, in quanto l'intero codice del sito web è pre-compilato in pochi file *dll* (spesso in un unico file) gestiti da un Server Web.



Microsoft
ASP.net



ASP Active
Server Pages

	Microsoft ASP.net	php	ASP Active Server Pages
License	Open source	Open source	Some unofficial open source projects
Programming type	Compiled	Scripting	Scripting
Development languages	VB.NET C# / C#.NET F#	PHP	Visual Basic
Development tools	Visual Studio Mono	<u>Netbeans</u> <u>PhpStorm</u> Any other text editor	Notepad++ Any other text editor
Server Operating system	Windows	Linux Windows (limited)	Windows
Security	Similar, developers have plenty of options on both – although .NET has stronger security at its core.		Weak
Update frequency	Every 2 months	Monthly	Never
Cost	Free, but advanced tools can be costly	Free	Free