

Dreischichtarchitektur: MVC

MVC

MODEL

"Intelligenz"
(z.B. Anwendungslogik);
Organisiert Daten (z.B.
aus/in Datenbank)

VIEW

Organisiert alle
Ausgaben (inkl.
Layout);
Templateverwaltung

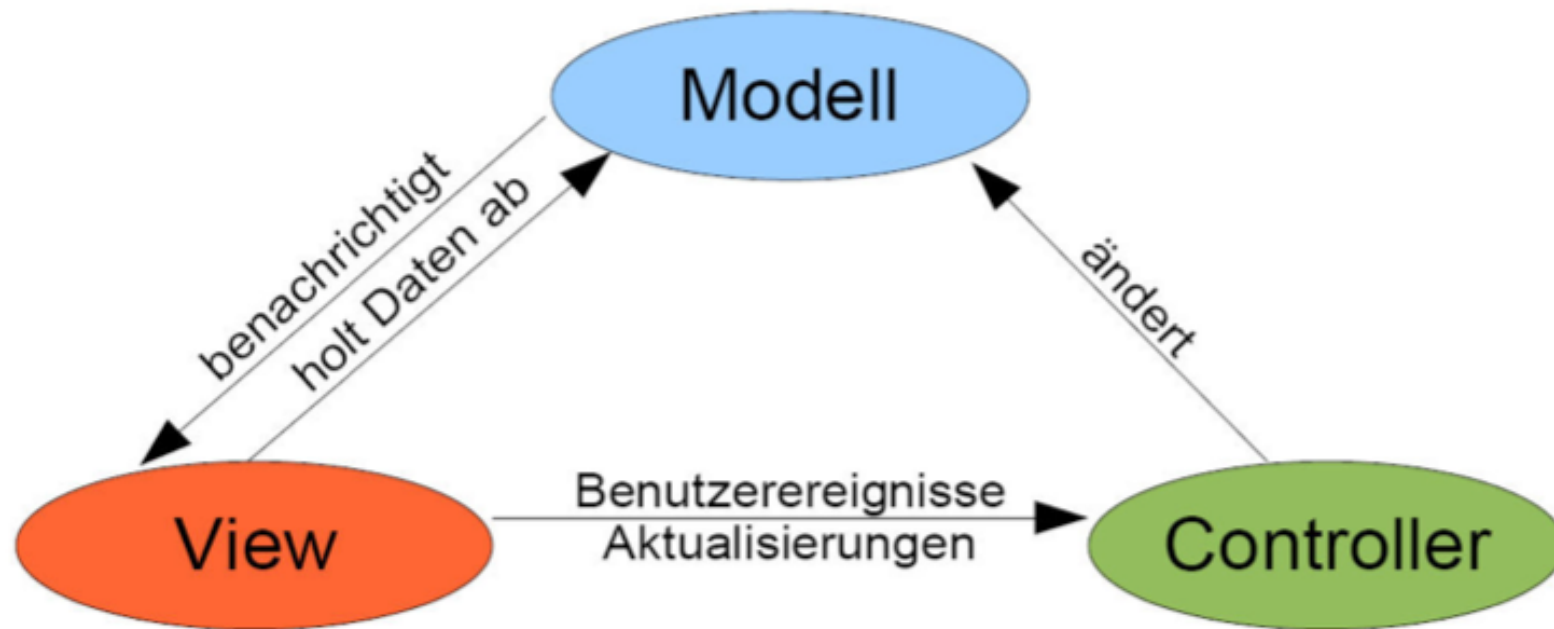
CONTROLLER

Steuert Anwendung:
Verwaltet Eingaben /
Benutzerinteraktion;
leitet entsprechende
Daten an Model weiter

Beispiel EVA-Prinzip:



MVC



https://www.matse.rz.rwth-aachen.de/dienste/public/show_document.php?id=7141

Warum MVC?

Einzelne Schichten unabhängig voneinander

- z.B. Präsentationsschicht kann völlig umgebaut werden, Programmlogik muss nicht verändert werden.
- z.B. Programmlogik kann erneuert werden, Darstellung bleibt erhalten
- z.B. Änderung der Programmiersprache; Darstellung bleibt erhalten

Entwicklungsaspekt, z.B.

Grafische Oberfläche kann völlig unabhängig von Programmlogik entwickelt werden

Keine einheitlichen Implementierungsstandards

Unterschiedliche Realisierungen, z.B.

- Überprüfung von Dateneingaben: im View, im Model oder im Controller?
(z.B. Prüfung auf korrektes Format einer E-Mail-Adresse)
- Anwendungslogik: Nur im Model? Kann auch (teilweise) im View oder im Controller sinnvoll sein
- Datenausgabe: Ausschließlich im View oder auch im Model?
(z.B. Rundung von Zahlen aus Datenbank)

**→MVC ist als Konzept zu sehen,
nicht als eindeutiges Kochrezept!**

Links

MVC in PHP umsetzen (eine Möglichkeit, mit Templates): Tutorial unter <http://tutorials.lemme.at/mvc-mit-php/>

Lesenswerter Übersichtsartikel bei Wikipedia:
http://de.wikipedia.org/wiki/Model_View_Controller