

MVC

Mozart Violin Concerto vs. Model-View-Controller



DON BURKS

Head Instructor, Lighthouse Labs
Vancouver, BC

@don_burks

<https://donburks.com>

don@lighthouselabs.ca

Also Don Burks...

BMA Musical Arts (Performance)

Columbus State University

1996



COLUMBUS
STATE
UNIVERSITY



A close-up photograph of a hand holding a blue and black plush toy. The toy has a large, circular, brown, textured object attached to its side. The background is dark and out of focus.

Obligatory
GIF

And now, for some **culture**.

VIOLINO PRINCIPALE. 3

The image shows a page of musical notation for the Violino Principale. The page is titled "VIOLINO PRINCIPALE." and has the number "3" in the top right corner. The music is written on six staves. The first staff begins with a treble clef and a key signature of one flat. The second staff has a "sol" marking above it. The third staff has a "D" marking above it and a "dolce" marking below it. The fourth staff has a "dolce" marking below it. The fifth staff has an "E" marking above it and a "p" marking below it. The sixth staff has a "dim" marking below it, a "p" marking below it, and a "f" marking below it. The music consists of various rhythmic patterns, including eighth and sixteenth notes, and rests.



OMG

WHO THE HELL CARES?

We should
care.

Because
musicians and
developers have
to **think** similarly.

Users care.

Because both
software and
music are a **craft**.

You should
care.

Because
approach is
everything.

I should
care.

Because I have to
give a **talk** on it.

Wolfgang Amadeus Mozart

1756 - 1791



A few relevant numbers

By the age of **36**, Mozart's body of work included...

41 Symphonies

27 Piano Concertos

5 Violin Concertos

27 Concert Arias

23 String Quartets

4 Horn Concertos

18 Masses

22 Operas

I'm 43, and I managed to put on pants.

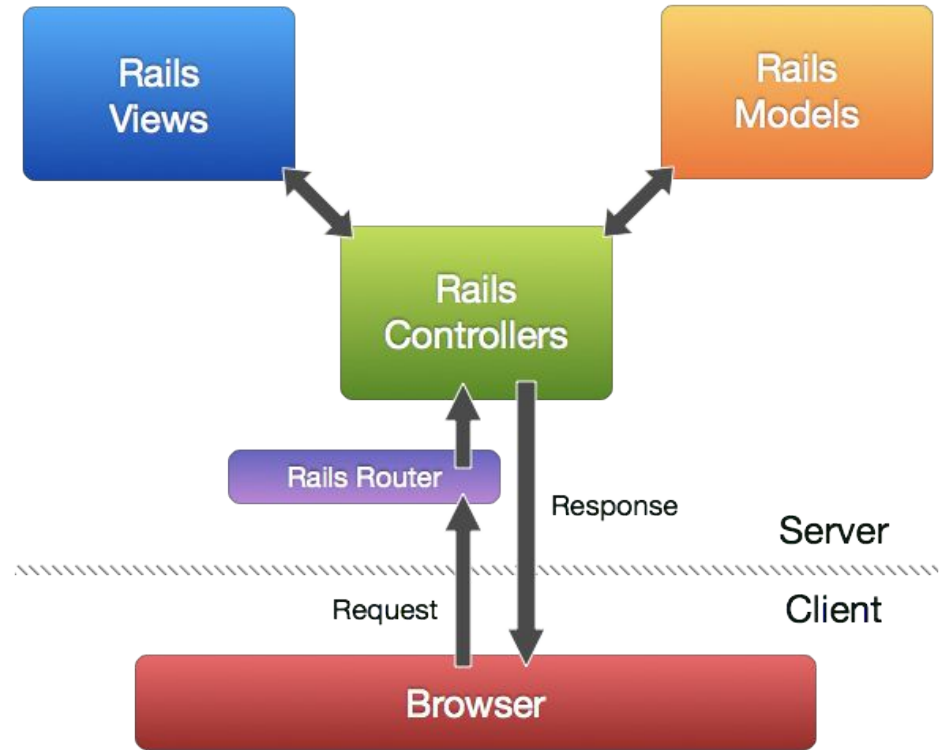


But what does
Mozart have
to do with code?

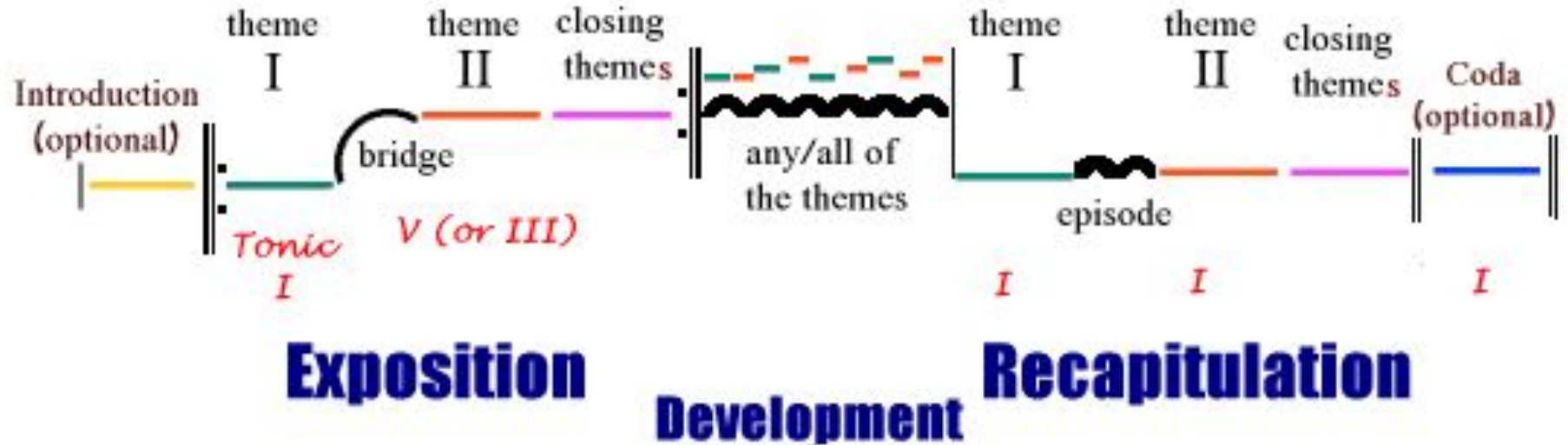


Structure

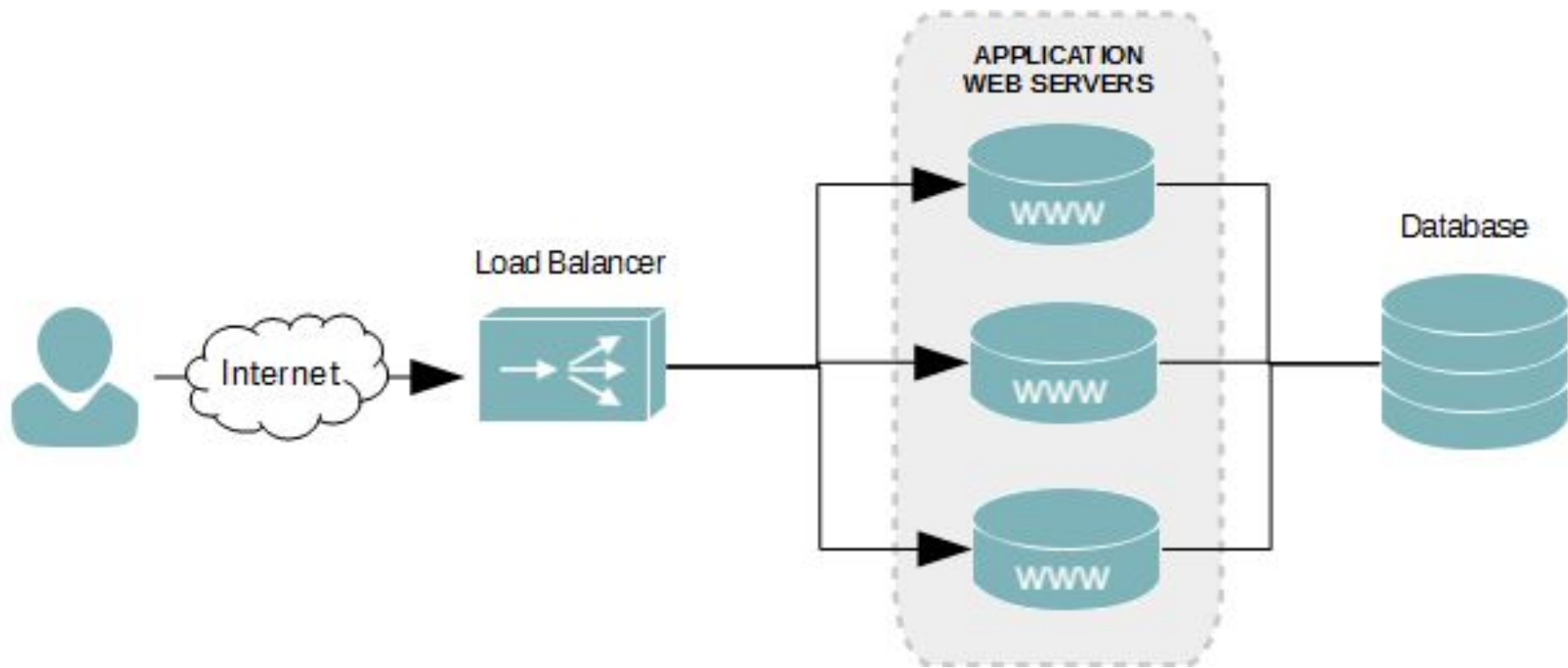
Apps have structure



Sonata-Allegro form

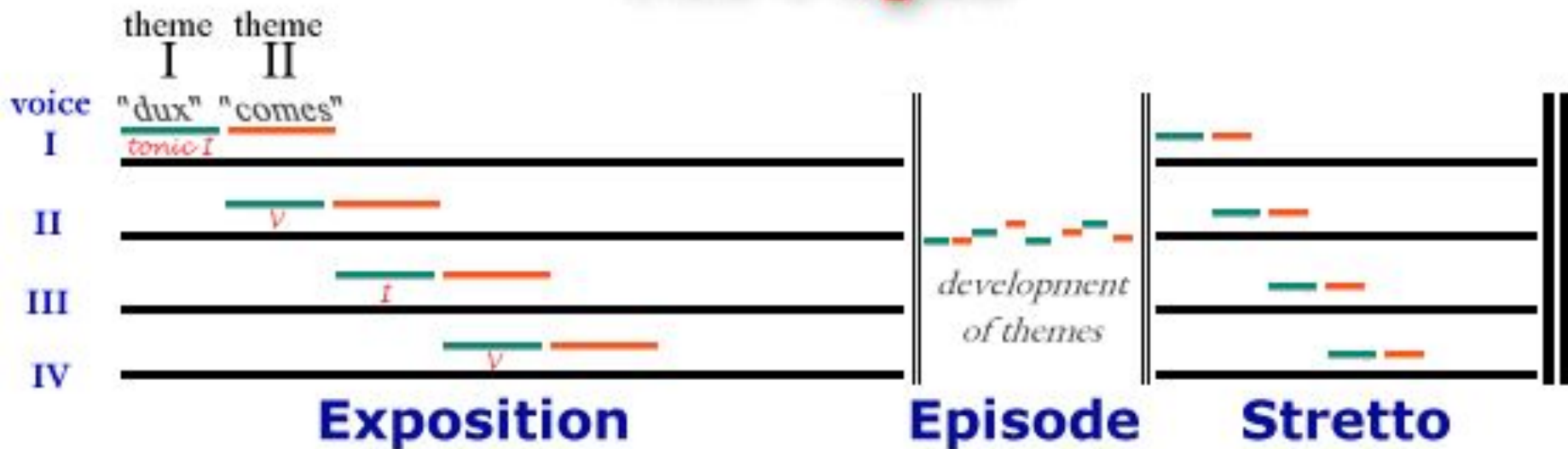


A Violin Concerto has **structure**, too.



We envision our applications as whole systems...

The Fugue



Composers have been doing this for centuries.

Approach

Horizontal

Lateral Thinking

Edward de Bono (1967)

Lateral thinking is solving problems through an indirect and creative approach, using reasoning that is not immediately obvious and involving ideas that may not be obtainable by using only traditional step-by-step logic.

★ [Wikipedia](#)

Horizontal Thinking in Musicians

Allegro moderato

Tutti



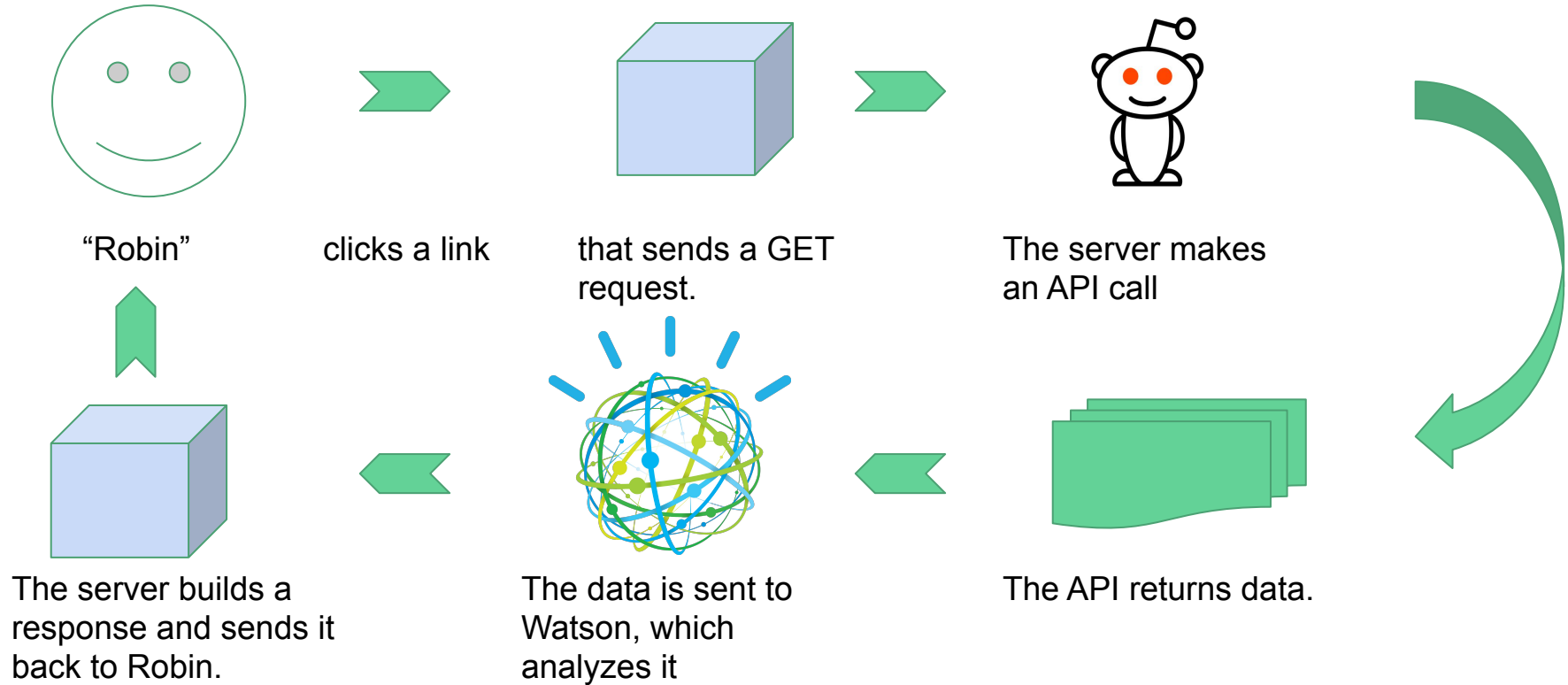
Melody moves the music forward

The image displays two staves of music. The first staff is in 3/4 time, starting with a treble clef and a key signature of one flat. It begins with a mezzo-forte (*mf*) dynamic and features a melodic line with eighth and quarter notes. Two green curved arrows point from the first two measures to the third and fourth measures, indicating a forward motion. A crescendo hairpin spans from the fourth measure to the end of the staff, where the dynamic reaches forte (*f*). The second staff, starting at measure 9, continues the melodic line with dotted notes and rests, marked with a 'V' above the first measure. The music concludes with a final melodic phrase.

“Algorithms are the melodies
of an application.”

- *A wise, humble man*

Horizontal Thinking in Developers



FizzBuzz

Great example!

Make a list of numbers from 1 - 100.

For each number:

Check to see if it is evenly divisible by 3. If yes, print “Fizz”.

Check to see if it is evenly divisible by 5. If yes, print “Buzz”.

Check to see if it is evenly divisible by both 3 **and** 5. If yes, print “FizzBuzz.”

Otherwise, just print the number.

Our code moves from
one step to another.
The thinking that fuels
our solutions is **linear**,
lateral, **horizontal**.

Vertical



Ember



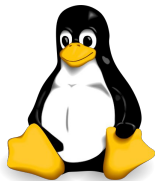
Ruby on Rails



PostgreSQL



Docker



Linux

We always call it a “**stack**”, don’t we?

We have to code vertically as well as horizontally

Web applications, desktop applications, mobile apps, all have a particular necessity for vertical elements. Included, but not limited to:

- AJAX
- APIs
- Isomorphism
- Client-side caching
- Proxies / VPNs
- Deployments
- PWA

Our challenge...

...is to have our work be
ignored.

Our horizontal problem-solving and vertical architecture should be transparent to our users.

Musicians

Have it **just** as hard





Do the beats align with the accompaniment?

The image shows a musical score with two staves. The top staff is marked *dolce* and contains a melodic line with various ornaments and slurs. The bottom staff is marked *p* and contains a harmonic accompaniment with long slurs. A green oval highlights a specific note in the top staff, which is a quarter note on the second line of the staff (F4).

As a musician in an ensemble, am I in tune?

“In tune,
On time.”

But can we apply that **mentality** to development?



**YES WE
CAN.**

A good user
experience is one
where the app runs...

“In tune, on time.”

Musicians vs. Developers

Musicians tell a story.

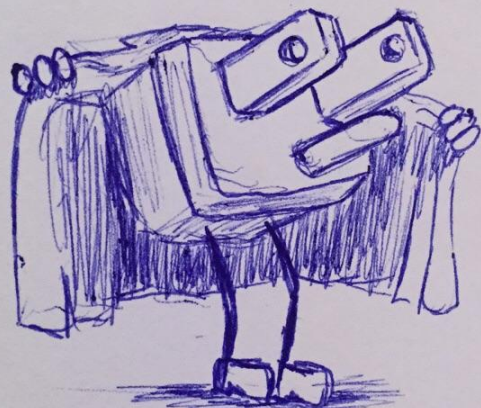
They decorate time instead of space.

A performer must string together notes into melodies, stacked on top of harmonies.

Developers offer a solution.

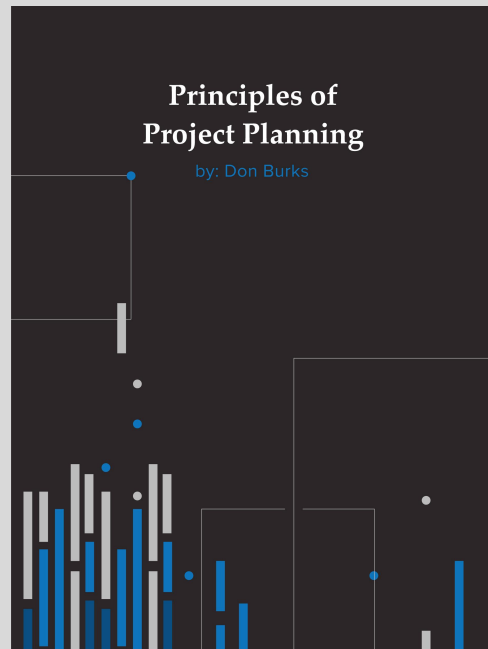
They give access to data, giving benefit through implementation.

An algorithm, implemented into an architecture, creates a product.



Shameless plug

BUY MY BOOK!



gum.co/project_planning

Thank you.

@don_burks

<https://donburks.com>

don@lighthouselabs.ca