



The title "DRAKARNAGE" is written in large, stylized, yellow-gold letters with a wood-grain texture. Below it, "VIKING WRESTLING!" is written in smaller, white letters with a stone or ice texture. The background features a grey gradient with a wooden spike and lightning bolt effect.



SLAPPING DATA FROM WEB SERVERS RIGHT INTO YOUR GPU



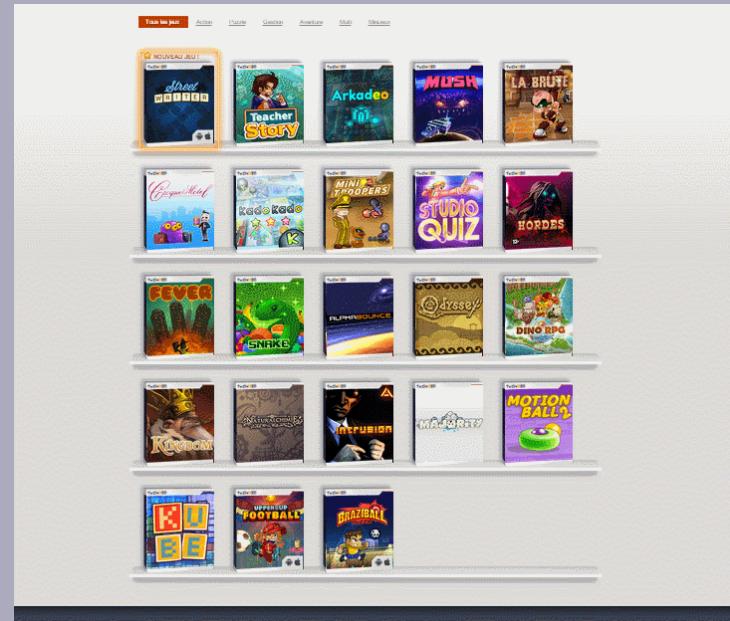
Motion Twin

Free 2 play pioneer

Haxe motherhouse

Game Jam addicts

Very small, Bossless and Self owned





David Elahee

Gaming industry veteran

Game Developer

GPU guy @motontwin

Haxe passionate



A good (solid) example of complex data flow...

Load Balancing

Service dispatch

Customising data end point

Prototyping

MOTION TWIN

Data repositories !

MySql

Redis

Mongo

Tora Share



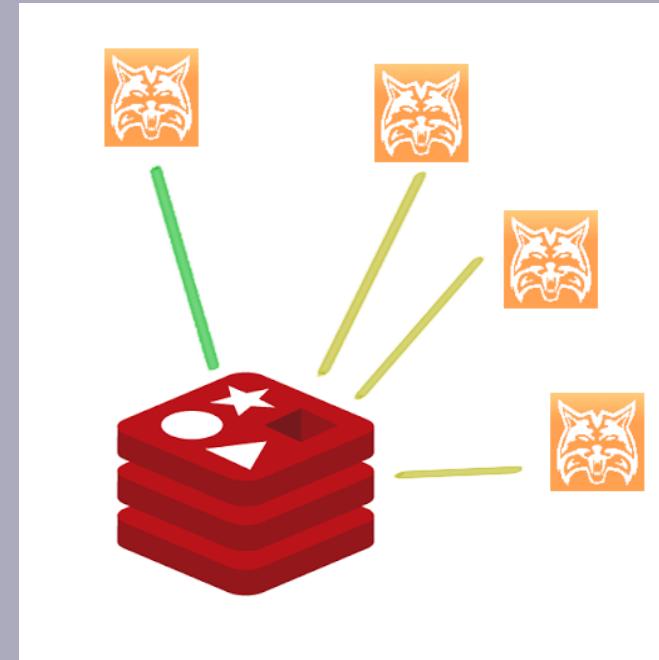


ex : Redis (Adv Key Values)

push pop

short lived data

sharing small chunks



MOTION TWIN

ex : Mongo (BSON documents)

fire and forget documents

incremental revisions

nice memory policy

complementary with MySQL





Right Storage for the right data !

Cost scaling

Easier to support

Many users, few servers



load balancing...

Tora queue

Keepalived

Dynamically pooled weighted servers

CDN





Failures...

External monitors

Solid community

Using the right service for the right data

DIY solutions



End points

Twinoid : social network

Facebook

Android

iOS



or...in your face...



End points...

Thanks to Haxe / OpenFL / NME

Same code for Web & Mobile

Minimalist proprietary extensions

Low maintenance cost





It is working !

Twinoid Mobile

Street Writer

will get in production for

Monster hotel, Rock Faller Journey, Drakarnage



Low maintenance cost ?

Game prototype in AIR (live reload, fast compile)

First data sets in Web/Flash/MySQL

Delivered to mobile (mostly) seamlessly...

Re routed to more specific data store transparently



What to do of it !

Games !

Viking Suplexes !

Moarr games fasterrr

Or apps or websites !





Framework : haxe

haxe dispatch / web service authoring

haxe.template + templo + hss / content templating - hot reloading

.po files / lower translation costs

hscript / artists tools - hot reloading



Framework : h3d !

Heaps twin brother (same genetics, grew up distorted ...)

Focused toward low end devices

Renders to opengl or stage3d with minimal efforts

State of the art 3d and 2d : batches, shader, skinning, morph etc...



Framework : h2d

sub part of h3d

very fast and auto/customizable batching (sprite & text)

advanced shader control



Framework : h2d

Tools to fit data for render timings

scout integration

rely heavily on reflection and hscript

custom shaders





But of course you all know that...

Let me show you how to combine all this !

MOTIONTWIN

render target



This is an HTML/Css page...

live edited
3d scene



h2d.comps

just like HTML/CSS without having to agree for standards

```
<body>
  <style>
    #main {
    }
  </style>
  <div id="bob" class="alice">
    ...
  </div>
</body>
```



h2d.comps

```
<style>
    viking{
        dock:bottom;
    }

    label#Foo{
        text-color-transform: hue(200) sat(-50); //deepnight trick
        height: 30x; // (integer pixel form our custom pixel ratio approximating of the 480x320 from current screen)
        blend-mode: add; //at least...
        filter:false; //here i come pixel perfect art!
    }
</style>
```



h2d.comps

Thanks to Nicolas C...It also sort of also has jquery...

```
<body>
  <style>
    ...
  </style>
  <viking id="ragnar" /> <!-- I can now hook this and make something tricky that can't be done with html-->
  <purchase />
  <script>
    $('purchase').show();
  </script>
</body>
```



html + h2d = comps !

same flow as regular HTML/H(c)SS templating

same flow as all our data !

finer control

pixel perfect ui's



production grease

local templating

application soft update

hot fixing

scalable ui is easy in html/css

MOTION TWIN

Prototyping

fast iteration

no compositing issues

finer fill rate control

easy to refine structure





why reinvent the wheel ?

use the correct tooling for each data

pool ressources

backward compatible flow

modern data manipulation



why reinvent the wheel - 2 ?

drakarnage is 2 people : hiko & myself & the motion twin team...

using standard services allows better tooling

using custom tools allows finer control, seamless collaboration



Questions ?

Thank You !

David Elahee @blackmagic_mt

