

ENTROPY WINS (HTTPS://WWW.ENTROPYWINS.WTF/BLOG/)

A blog on Software Architecture, Design and Craftsmanship

Clean Code (https://www.entropywins.wtf/blog/tag/clean-code/)

Craftsmanship (https://www.entropywins.wtf/blog/tag/software-craftsmanship/)

Architecture (https://www.entropywins.wtf/blog/tag/software-architecture/)

Testing (https://www.entropywins.wtf/blog/tag/testing/)

PHP (https://www.entropywins.wtf/blog/tag/php/)

Open Source (https://www.entropywins.wtf/blog/tag/open-source/)

MediaWiki (https://www.entropywins.wtf/blog/tag/mediawiki/)

Website home (https://www.entropywins.wtf/)

Professional.Wiki (https://professional.wiki/)

 [FOLLOW ME ON TWITTER \(HTTPS://TWITTER.COM/JEROENDEDAUW\)](https://twitter.com/JEROENDEDAUW)

 [FOLLOW ME ON GITHUB \(HTTPS://GITHUB.COM/JEROENDEDAUW\)](https://github.com/JEROENDEDAUW)

[RSS \(HTTP://WWW.BN2VS.COM/BLOG/FEED/\)](http://www.bn2vs.com/blog/feed/)

[ATOM \(HTTP://WWW.BN2VS.COM/BLOG/FEED/ATOM/\)](http://www.bn2vs.com/blog/feed/atom/)

[PRIVACY POLICY \(HTTPS://WWW.ENTROPYWINS.WTF/PRIVACY-POLICY\)](https://www.entropywins.wtf/privacy-policy)

Clean Architecture + Bounded Contexts

🕒 2018-08-14 👤 Jeroen (<https://www.entropywins.wtf/blog/author/jeroen/>) 💬 5 Comments
(<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/#comments>).

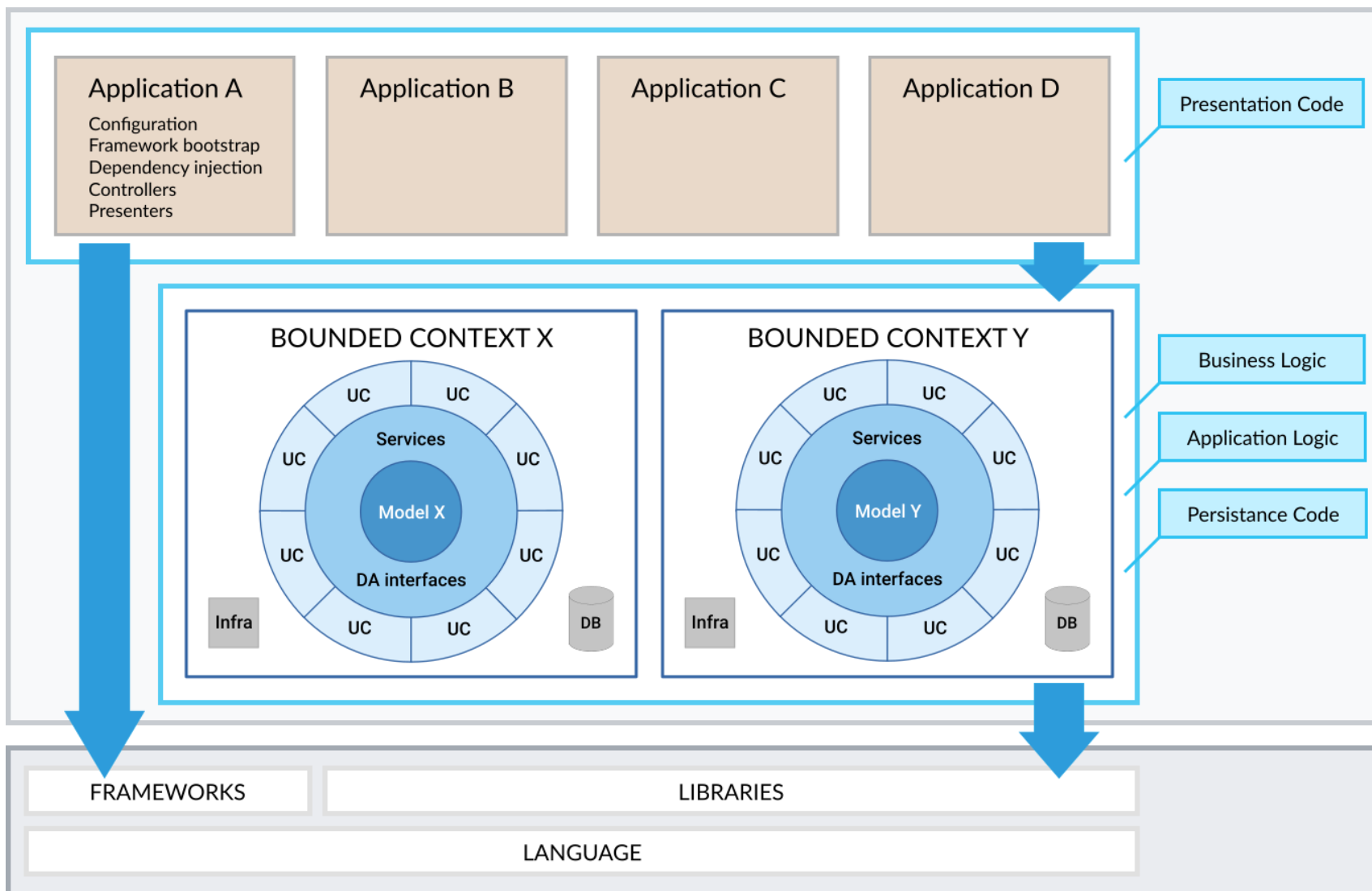
In this follow-up to [Implementing the Clean Architecture](https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/) (<https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/>), I introduce you to a combination of The Clean Architecture and the strategic DDD pattern known as Bounded Contexts.

At [Wikimedia Deutschland](https://wikimedia.de/en/) (<https://wikimedia.de/en/>), we use this combination of The Clean Architecture and [Bounded Contexts](https://martinfowler.com/bliki/BoundedContext.html) (<https://martinfowler.com/bliki/BoundedContext.html>) for our fundraising applications. In this post I describe the structure we have and the architectural rules we follow in the abstract. For the story on how we got to this point and a more concrete description, see my post [Bounded Contexts in the Wikimedia Fundraising Software](https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/) (<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>). In that post and at the end of this one I link you to a real-world codebase that follows the abstract rules described in this post.

If you are not yet familiar with The Clean Architecture, please first read [Implementing the Clean Architecture](https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/) (<https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/>).

Clean Architecture + Bounded Contexts

Diagram by Jeroen De Dauw, Charlie Kritschmar, Jan Dittrich and Hanna Petruschat. Click image to enlarge



(https://www.entropywins.wtf/blog/wp-content/uploads/2018/08/CleanArchitecture_DomainDrivenDesign.png).

In the top layer of the diagram we have applications. These can be web applications, they can be console applications, they can be monoliths, they can be microservices, etc. Each application has presentation code which in bigger applications tends to reside in a decoupled presentation layer using patterns such as presenters. All applications also somehow construct the dependency graph they need, perhaps using a Dependency Injection Container or set of factories. Often this involves reading configuration from somewhere. The applications contain ALL framework binding, hence they are the place where you will find the Controllers if you are using a typical web framework.

Since the applications are in the top layer, and dependencies can only go down, no code outside of the applications is allowed to depend on code in the applications. That means there is 0 binding to mechanisms such as frameworks and presentation code outside of the applications.

In the second layer we have the Bounded Contexts. Ideally one Bounded Context per subdomain. At the core of each BC we have the Domain Model and Domain Services, containing the business logic part of the subdomain. Dependencies can only point inwards, so the Domain model which is at the center cannot depend on anything more to the outside. Around the Domain Model are the Domain Services. These include interfaces for persistence services such as Repositories. The UseCases form the final ring. They can use both the Domain Model and the Domain Services. They also form a boundary around the two, meaning that no code outside of the Bounded Context is allowed to talk to the Domain Model or Domain Services.

The Bounded Contexts include their own Persistence Layer. The Persistence Layer can use a relational database, files on the file system, a remote web API, a combination of these, etc. It has implementations of domain services such as Repositories which are used by the UseCases. These implementations are the only thing that is allowed to talk to and know about the low-level aspects of the Persistence Layer. The only things that can use these service implementations are other Domain Services and the UseCases.

The UseCases, including their Request Models and Response Models, form the public interface of the Bounded Context. This means that there is 0 binding to the persistence mechanisms outside of the Bounded Context. It also means that the code responsible for the domain logic cannot be directly accessed elsewhere, such as in the presentation layer of an application.

The applications and Bounded Contexts contain all the domain specific code. This code can make use of libraries and of course the runtime (ie PHP) itself.

As examples of Bounded Contexts following this approach, see the Donation Context (<https://github.com/wmde/fundraising-donations>) and Membership Context (<https://github.com/wmde/fundraising-memberships>). For an application following this architecture, see the FundraisingFrontend (<https://github.com/wmde/FundraisingFrontend>), which uses both the

Donation Context and Membership Context. Both these contexts are also used by another application the code of which sadly enough is not currently public. You can also read the stories of [how we rewrote the FundraisingFrontend to use the Clean Architecture](https://www.entropywins.wtf/blog/2016/11/24/rewriting-the-wikimedia-deutschland-fundraising/) (<https://www.entropywins.wtf/blog/2016/11/24/rewriting-the-wikimedia-deutschland-fundraising/>) and [how we refactored towards Bounded Contexts](https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/) (<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>).

FURTHER READING

Sign up below to receive news on my upcoming Clean Architecture book, including a discount:

Your email address

SUBSCRIBE

If you are not yet familiar with Bounded Contexts or how to design them well, I recommend reading [Domain-Driven Design Distilled](https://www.goodreads.com/book/show/28602719-domain-driven-design-distilled) (<https://www.goodreads.com/book/show/28602719-domain-driven-design-distilled>).



Jeroen De Dauw

@JeroenDeDauw

Blogged: Clean Architecture + Bounded Contexts

[entropywins.wtf/blog/2018/08/14/rewriting-the-wikimedia-deutschland-fundraising/](https://www.entropywins.wtf/blog/2018/08/14/rewriting-the-wikimedia-deutschland-fundraising/) #CleanCode #CleanArchitecture #DDDesign

♡ 2

9:13 AM - Aug 16, 2018

 See Jeroen De Dauw's other Tweets

>

 Post Views: 12,930

SHARE THIS:

Twitter (<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/?share=twitter&nb=1>)

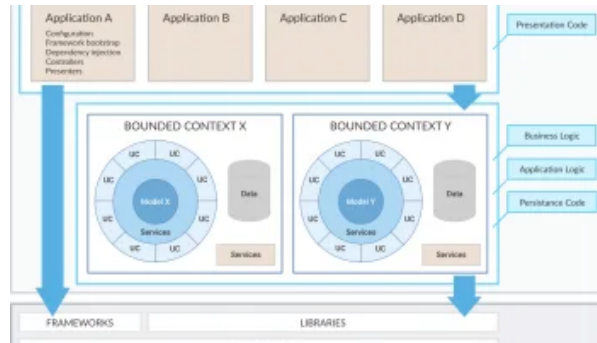
Facebook (<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/?share=facebook&nb=1>)

Pocket (<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/?share=pocket&nb=1>)

Reddit (<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/?share=reddit&nb=1>)

More

RELATED



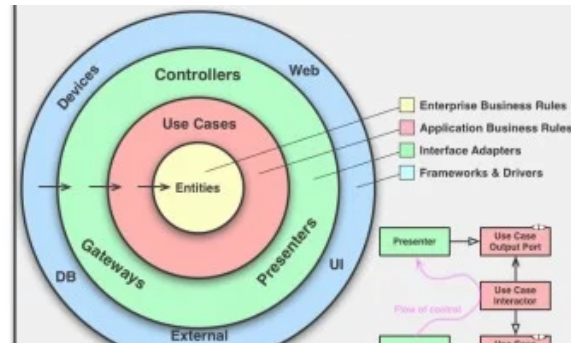
(<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>)

Bounded Contexts in the Wikimedia Fundraising Software

(<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>)

2018-08-14

In "Programming"



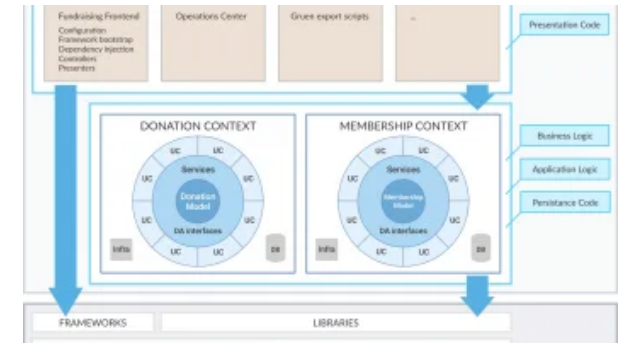
(<https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/>)

Implementing the Clean Architecture

(<https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/>)

2016-11-24

In "Programming"



(<https://www.entropywins.wtf/blog/2018/09/09/clean-architecture-bounded-contexts-diagram/>)

Clean Architecture + Bounded Contexts diagram

(<https://www.entropywins.wtf/blog/2018/09/09/clean-architecture-bounded-contexts-diagram/>)

2018-09-09

In "Programming"

← Bounded Contexts in the Wikimedia Fundraising Software
(<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>)

Base Libraries Should be Stable →
(<https://www.entropywins.wtf/blog/2018/08/30/base-libraries-should-be-stable/>)

5 thoughts on “Clean Architecture + Bounded Contexts”

Pingback: Bounded Contexts in the Wikimedia Fundraising Software – Entropy Wins
(<https://www.entropywins.wtf/blog/2018/08/14/bounded-contexts-in-the-wikimedia-fundraising-software/>).

Pingback: Implementing the Clean Architecture – Entropy Wins
(<https://www.entropywins.wtf/blog/2016/11/24/implementing-the-clean-architecture/>).

Pingback: Clean Architecture: UseCase tests – Entropy Wins (<https://www.entropywins.wtf/blog/2018/08/01/clean-architecture-usecase-tests/>).

Pingback: Clean Architecture diagrams – Entropy Wins (<https://www.entropywins.wtf/blog/2016/09/09/clean-architecture-diagram/>).

Pingback: Clean Architecture + Bounded Contexts diagram – Entropy Wins
(<https://www.entropywins.wtf/blog/2018/09/09/clean-architecture-bounded-contexts-diagram/>).

LEAVE A REPLY

Enter your comment here...

This site uses Akismet to reduce spam. [Learn how your comment data is processed \(https://akismet.com/privacy/\)](https://akismet.com/privacy/).

Enjoying my blog or open source projects? Consider [giving something small back!](https://github.com/sponsors/JeroenDeDauw)
(<https://github.com/sponsors/JeroenDeDauw>). <3

Search ...

NEWSLETTER

Sign up below to receive news on my upcoming Clean Architecture book, including a discount:

Your email address

SUBSCRIBE

RECENT POSTS

[New MediaWiki blog \(https://www.entropywins.wtf/blog/2019/10/26/new-mediawiki-blog/\)](https://www.entropywins.wtf/blog/2019/10/26/new-mediawiki-blog/).

[Applications as Frameworks \(https://www.entropywins.wtf/blog/2019/02/28/applications-as-frameworks/\)](https://www.entropywins.wtf/blog/2019/02/28/applications-as-frameworks/).

[Readable Functions: Guard Clause \(https://www.entropywins.wtf/blog/2019/01/14/readable-functions-guard-clause/\)](https://www.entropywins.wtf/blog/2019/01/14/readable-functions-guard-clause/).

My year in books (<https://www.entropywins.wtf/blog/2019/01/03/my-year-in-books-4/>).

Readable Functions: Do One Thing (<https://www.entropywins.wtf/blog/2018/10/30/readable-functions-do-one-thing/>).

PHP Typed Properties (<https://www.entropywins.wtf/blog/2018/10/28/php-typed-properties/>).

Readable Functions: Minimize State (<https://www.entropywins.wtf/blog/2018/10/24/readable-functions-minimize-state/>).

Clean Architecture + Bounded Contexts diagram (<https://www.entropywins.wtf/blog/2018/09/09/clean-architecture-bounded-contexts-diagram/>).

Base Libraries Should be Stable (<https://www.entropywins.wtf/blog/2018/08/30/base-libraries-should-be-stable/>).

Clean Architecture + Bounded Contexts (<https://www.entropywins.wtf/blog/2018/08/14/clean-architecture-bounded-contexts/>).

ARCHIVES

October 2019 (<https://www.entropywins.wtf/blog/2019/10/>).

February 2019 (<https://www.entropywins.wtf/blog/2019/02/>).

January 2019 (<https://www.entropywins.wtf/blog/2019/01/>).

October 2018 (<https://www.entropywins.wtf/blog/2018/10/>).

September 2018 (<https://www.entropywins.wtf/blog/2018/09/>).

August 2018 (<https://www.entropywins.wtf/blog/2018/08/>).

May 2018 (<https://www.entropywins.wtf/blog/2018/05/>).

February 2018 (<https://www.entropywins.wtf/blog/2018/02/>).

January 2018 (<https://www.entropywins.wtf/blog/2018/01/>).

October 2017 (<https://www.entropywins.wtf/blog/2017/10/>)

September 2017 (<https://www.entropywins.wtf/blog/2017/09/>)

June 2017 (<https://www.entropywins.wtf/blog/2017/06/>)

May 2017 (<https://www.entropywins.wtf/blog/2017/05/>)

April 2017 (<https://www.entropywins.wtf/blog/2017/04/>)

February 2017 (<https://www.entropywins.wtf/blog/2017/02/>)

January 2017 (<https://www.entropywins.wtf/blog/2017/01/>)

December 2016 (<https://www.entropywins.wtf/blog/2016/12/>)

November 2016 (<https://www.entropywins.wtf/blog/2016/11/>)

October 2016 (<https://www.entropywins.wtf/blog/2016/10/>)

September 2016 (<https://www.entropywins.wtf/blog/2016/09/>)

August 2016 (<https://www.entropywins.wtf/blog/2016/08/>)

July 2016 (<https://www.entropywins.wtf/blog/2016/07/>)

June 2016 (<https://www.entropywins.wtf/blog/2016/06/>)

May 2016 (<https://www.entropywins.wtf/blog/2016/05/>)

April 2016 (<https://www.entropywins.wtf/blog/2016/04/>)

March 2016 (<https://www.entropywins.wtf/blog/2016/03/>)

February 2016 (<https://www.entropywins.wtf/blog/2016/02/>)

January 2016 (<https://www.entropywins.wtf/blog/2016/01/>)

[December 2015 \(https://www.entropywins.wtf/blog/2015/12/\)](https://www.entropywins.wtf/blog/2015/12/)

[November 2015 \(https://www.entropywins.wtf/blog/2015/11/\)](https://www.entropywins.wtf/blog/2015/11/)

[October 2015 \(https://www.entropywins.wtf/blog/2015/10/\)](https://www.entropywins.wtf/blog/2015/10/)

[September 2015 \(https://www.entropywins.wtf/blog/2015/09/\)](https://www.entropywins.wtf/blog/2015/09/)

[August 2015 \(https://www.entropywins.wtf/blog/2015/08/\)](https://www.entropywins.wtf/blog/2015/08/)

[June 2015 \(https://www.entropywins.wtf/blog/2015/06/\)](https://www.entropywins.wtf/blog/2015/06/)

[February 2015 \(https://www.entropywins.wtf/blog/2015/02/\)](https://www.entropywins.wtf/blog/2015/02/)

[September 2014 \(https://www.entropywins.wtf/blog/2014/09/\)](https://www.entropywins.wtf/blog/2014/09/)

[August 2014 \(https://www.entropywins.wtf/blog/2014/08/\)](https://www.entropywins.wtf/blog/2014/08/)

[July 2014 \(https://www.entropywins.wtf/blog/2014/07/\)](https://www.entropywins.wtf/blog/2014/07/)

[May 2014 \(https://www.entropywins.wtf/blog/2014/05/\)](https://www.entropywins.wtf/blog/2014/05/)

[April 2014 \(https://www.entropywins.wtf/blog/2014/04/\)](https://www.entropywins.wtf/blog/2014/04/)

[March 2014 \(https://www.entropywins.wtf/blog/2014/03/\)](https://www.entropywins.wtf/blog/2014/03/)

[February 2014 \(https://www.entropywins.wtf/blog/2014/02/\)](https://www.entropywins.wtf/blog/2014/02/)

[January 2014 \(https://www.entropywins.wtf/blog/2014/01/\)](https://www.entropywins.wtf/blog/2014/01/)

[December 2013 \(https://www.entropywins.wtf/blog/2013/12/\)](https://www.entropywins.wtf/blog/2013/12/)

[November 2013 \(https://www.entropywins.wtf/blog/2013/11/\)](https://www.entropywins.wtf/blog/2013/11/)

[October 2013 \(https://www.entropywins.wtf/blog/2013/10/\)](https://www.entropywins.wtf/blog/2013/10/)

[September 2013 \(https://www.entropywins.wtf/blog/2013/09/\)](https://www.entropywins.wtf/blog/2013/09/)

[July 2013 \(https://www.entropywins.wtf/blog/2013/07/\)](https://www.entropywins.wtf/blog/2013/07/)

[June 2013 \(https://www.entropywins.wtf/blog/2013/06/\)](https://www.entropywins.wtf/blog/2013/06/)

[March 2012 \(https://www.entropywins.wtf/blog/2012/03/\)](https://www.entropywins.wtf/blog/2012/03/)

[December 2011 \(https://www.entropywins.wtf/blog/2011/12/\)](https://www.entropywins.wtf/blog/2011/12/)

[November 2011 \(https://www.entropywins.wtf/blog/2011/11/\)](https://www.entropywins.wtf/blog/2011/11/)

[September 2011 \(https://www.entropywins.wtf/blog/2011/09/\)](https://www.entropywins.wtf/blog/2011/09/)

[August 2011 \(https://www.entropywins.wtf/blog/2011/08/\)](https://www.entropywins.wtf/blog/2011/08/)

[July 2011 \(https://www.entropywins.wtf/blog/2011/07/\)](https://www.entropywins.wtf/blog/2011/07/)

[June 2011 \(https://www.entropywins.wtf/blog/2011/06/\)](https://www.entropywins.wtf/blog/2011/06/)

[May 2011 \(https://www.entropywins.wtf/blog/2011/05/\)](https://www.entropywins.wtf/blog/2011/05/)

[February 2011 \(https://www.entropywins.wtf/blog/2011/02/\)](https://www.entropywins.wtf/blog/2011/02/)

[January 2011 \(https://www.entropywins.wtf/blog/2011/01/\)](https://www.entropywins.wtf/blog/2011/01/)

[December 2010 \(https://www.entropywins.wtf/blog/2010/12/\)](https://www.entropywins.wtf/blog/2010/12/)

[October 2010 \(https://www.entropywins.wtf/blog/2010/10/\)](https://www.entropywins.wtf/blog/2010/10/)

[August 2010 \(https://www.entropywins.wtf/blog/2010/08/\)](https://www.entropywins.wtf/blog/2010/08/)

[July 2010 \(https://www.entropywins.wtf/blog/2010/07/\)](https://www.entropywins.wtf/blog/2010/07/)

[June 2010 \(https://www.entropywins.wtf/blog/2010/06/\)](https://www.entropywins.wtf/blog/2010/06/)

[May 2010 \(https://www.entropywins.wtf/blog/2010/05/\)](https://www.entropywins.wtf/blog/2010/05/)

[April 2010 \(https://www.entropywins.wtf/blog/2010/04/\)](https://www.entropywins.wtf/blog/2010/04/)

[March 2010 \(https://www.entropywins.wtf/blog/2010/03/\)](https://www.entropywins.wtf/blog/2010/03/)

[February 2010 \(https://www.entropywins.wtf/blog/2010/02/\)](https://www.entropywins.wtf/blog/2010/02/)

[January 2010 \(https://www.entropywins.wtf/blog/2010/01/\)](https://www.entropywins.wtf/blog/2010/01/)

[December 2009 \(https://www.entropywins.wtf/blog/2009/12/\)](https://www.entropywins.wtf/blog/2009/12/)

[November 2009 \(https://www.entropywins.wtf/blog/2009/11/\)](https://www.entropywins.wtf/blog/2009/11/)

[October 2009 \(https://www.entropywins.wtf/blog/2009/10/\)](https://www.entropywins.wtf/blog/2009/10/)

[September 2009 \(https://www.entropywins.wtf/blog/2009/09/\)](https://www.entropywins.wtf/blog/2009/09/)

[August 2009 \(https://www.entropywins.wtf/blog/2009/08/\)](https://www.entropywins.wtf/blog/2009/08/)

[July 2009 \(https://www.entropywins.wtf/blog/2009/07/\)](https://www.entropywins.wtf/blog/2009/07/)

[June 2009 \(https://www.entropywins.wtf/blog/2009/06/\)](https://www.entropywins.wtf/blog/2009/06/)

[May 2009 \(https://www.entropywins.wtf/blog/2009/05/\)](https://www.entropywins.wtf/blog/2009/05/)

[April 2009 \(https://www.entropywins.wtf/blog/2009/04/\)](https://www.entropywins.wtf/blog/2009/04/)