

# The Relevance of FOSS License Compatibility

Open Source Preservation Workshop, PREFORMA, Stockholm, April 7, 2016

Dr. Till Jaeger

Certified Copyright and Media Law Attorney

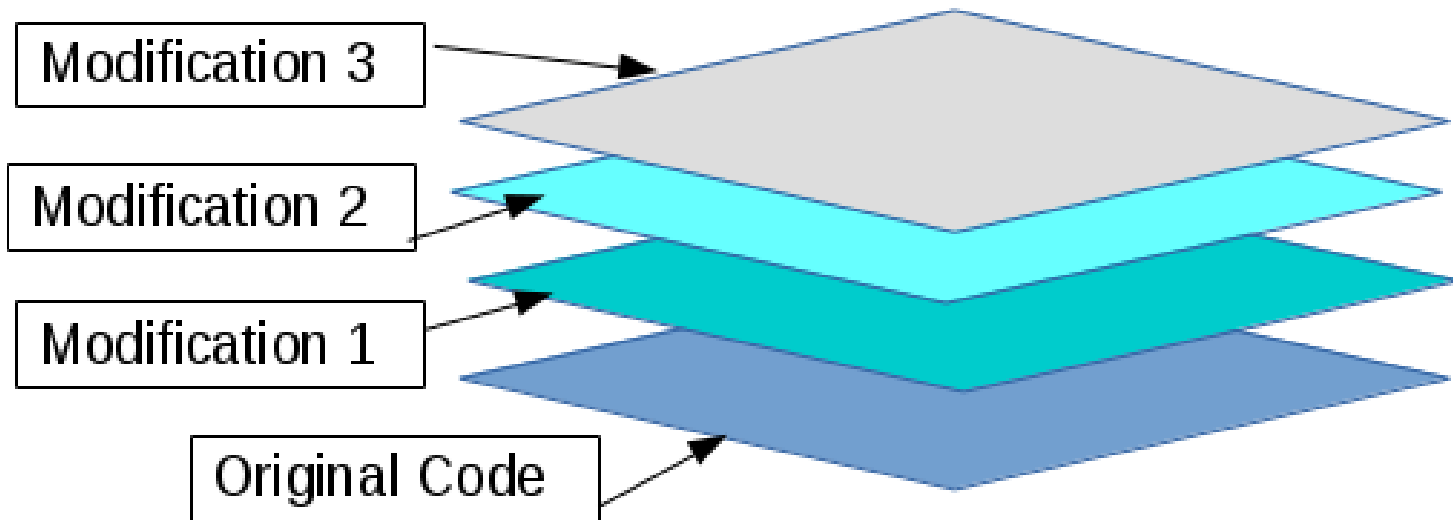
# Derivative Works

Modifications of preexisting software result in a copyright on the modified software (if modification is individual rather than trivial) - „derivative works“

The copyright holder of the modified work needs the permission of the copyright holder of the original work to use the modified work

The copyright holder of the original work can control if modifications are made and how to use them

# Copyright on Derivative Works



# Copyleft and License Categories

“Copyleft”: Obligation to the licensee to license derivative works under the original OSS license (in case of distribution)

“Legal hack“ of Richard Stallman to avoid that licensees do not provide the source code of their modifications and proprietary use of the modified software

The concept of Copyleft: using copyright to ensure free use of a software instead of collecting royalties

# Copyleft and License Categories

## Copyleft licenses

- Strong copyleft licenses: all modifications have to be licensed under the original license if distributed
- Weak copyleft licenses: some modifications have to be licensed under the original license if distributed

Non-Copyleft licenses (permissive licenses): derivative works may be licensed under different license conditions, even proprietary

# Copyleft and License Categories

Licenses with strong Copyleft:

- GPL - GNU General Public License, versions 2 and 3
- AGPL - GNU Affero General Public License
- EPL – Eclipse Public License
- EUPL – European Union Public License

Licenses with weak Copyleft

- LGPL – GNU Lesser General Public License, versions 2, 2.1 and 3
- MPL – Mozilla Public License, versions 1, 1.1 and 2

# Copyleft and License Categories

Non-Copyleft licenses:

- BSD License (2-Clause, 3-Clause and 4-Clause)
- Apache License 2.0
- MIT License
- Academic Free License
- PHP License

...

# The Impact of Copyleft

The Copyleft is relevant for:

- Licensing your own developments/modifications
- Compatibility with other OSS-components and remixing code

Mostly a problem of copyright law: what is a “derivative work” and triggers Copyleft?

Currently no case law that involve GPL interpretation or concerning the concept of adaptation of software, but: VMware case currently at court (District Court of Hamburg)



# The Impact of Copyleft

Relevant for the question whether code from one Open Source component can be copied in another component

Relevant for the question whether Open Source components under various licenses can be combined together within one work

vp8\_cx\_iface.c  
3-Clause-BSD

```
if (!ctx->priv)
{
    struct vp8_codec_alg_priv *priv =
        (struct vp8_codec_alg_priv
         *)vpx_malloc(1, sizeof(*priv));

    if (!priv)
    {
        return VP8_CODEC_MEM_ERROR;
    }

    ctx->priv = (vp8_codec_priv_t *)priv;
    ctx->priv->init_flags = ctx->init_flags;

    if (ctx->config_enc)
    {
        /* Update the reference to the config
         structure to an
         * internal copy.
         */
        priv->cfg = *ctx->config_enc;
        ctx->config_enc = &priv->cfg;
    }

    priv->vp8_cfg = default_extractcfg;
    priv->vp8_cfg_pkt_list = &priv-
>pkt_list_head;

    priv->cx_data_sz = priv->cfg.g_w * priv-
>cfg.g_h * 3 / 2 * 2;

    if (priv->cx_data_sz < 32768) priv-
>cx_data_sz = 32768;

    priv->cx_data = malloc(priv->cx_data_sz);
}
```

getmsg.c  
GPL-2.0

```
#include <errno.h>
#include <stropts.h>

int
getmsg (fildes, ctlptr, dataptr, flagsp)
    int fildes;
    struct strbuf *ctlptr;
    struct strbuf *dataptr;
    int *flagsp;
{
    __set_errno (ENOSYS);
    return -1;
}

stub_warning (getmsg)
#include <stub-tag.h>
```

vp8\_cx\_iface.c  
3-Clause-BSD

```
if (!ctx->priv)
{
    struct vpx_codec_alg_priv *priv =
        (struct vpx_codec_alg_priv
         *)vpx_malloc(1, sizeof(*priv));

    if (!priv)
    {
        return VPX_CODEC_MEM_ERROR;
    }

    ctx->priv = (vpx_codec_priv_t *)priv;
    ctx->init_flags = ctx->init_flags;

    if (ctx->config_enc)
    {
        /* Update the reference to the config
        structure to an
        * internal copy.
        */
        priv->cfg = *ctx->config_enc;
        ctx->config_enc = &priv->cfg;
    }

    priv->vp8_cfg = default_extractcfg;
    priv->vp8_cfg.pkt_list = &priv->
    >pkt_list.head;

    priv->cx_data_sz = priv->cfg.g_w * priv->
    >cfg.g_h * 3 / 2 * 2;

    if (priv->cx_data_sz < 32768) priv->
    >cx_data_sz = 32768;

    priv->cx_data = malloc(priv->cx_data_sz);
}
```

getmsg.c  
GPL-2.0

```
#include <errno.h>
#include <stropts.h>

int
getmsg (files, ctiptr, dataptr, flagsp)
    int files;
    struct strbuf *ctiptr;
    struct strbuf *dataptr;
    int *flagsp;
{
    __set_errno (ENOSYS);
    return -1;
}

stub_warning (getmsg)
#include <stub-tag.h>
```



vp8\_cx\_iface.c  
3-Clause-BSD

```
if (!ctx->priv)
{
    struct vpx_codec_alg_priv *priv =
        (struct vpx_codec_alg_priv
         *)vpx_calloc(1, sizeof(*priv));

    if (!priv)
    {
        return VPX_CODEC_MEM_ERROR;
    }

    ctx->priv = (vpx_codec_priv_t *)priv;
    ctx->priv->init_flags = ctx->init_flags;

    if (ctx->config_enc)
    {
        /* Update the reference to the config
         structure to an
         * internal copy.
         */
        priv->cfg = *ctx->config_enc;
        ctx->config_enc = &priv->cfg;
    }

    priv->vp8_cfg = default_extractcfg;
    priv->vp8_cfg_pkt_list = &priv-
    >pkt_list.head;

    priv->cx_data_sz = priv->cfg.g_w * priv-
    >cfg.g_h * 3 / 2 * 2;

    if (priv->cx_data_sz < 32768) priv-
    >cx_data_sz = 32768;

    priv->cx_data = malloc(priv->cx_data_sz);
}
```



getmsg.c  
GPL-2.0

```
#include <errno.h>
#include <stropts.h>

int
getmsg (files, ctpr, dataptr, flagsp)
    int files;
    struct strbuf *ctpr;
    struct strbuf *dataptr;
    int *flagsp;
{
    __set_errno (ENOSYS);
    return -1;
}

if (ctx->config_enc)
{
    /* Update the reference to the config
     structure to an
     * internal copy.
     */
    priv->cfg = *ctx->config_enc;
    ctx->config_enc = &priv->cfg;
}

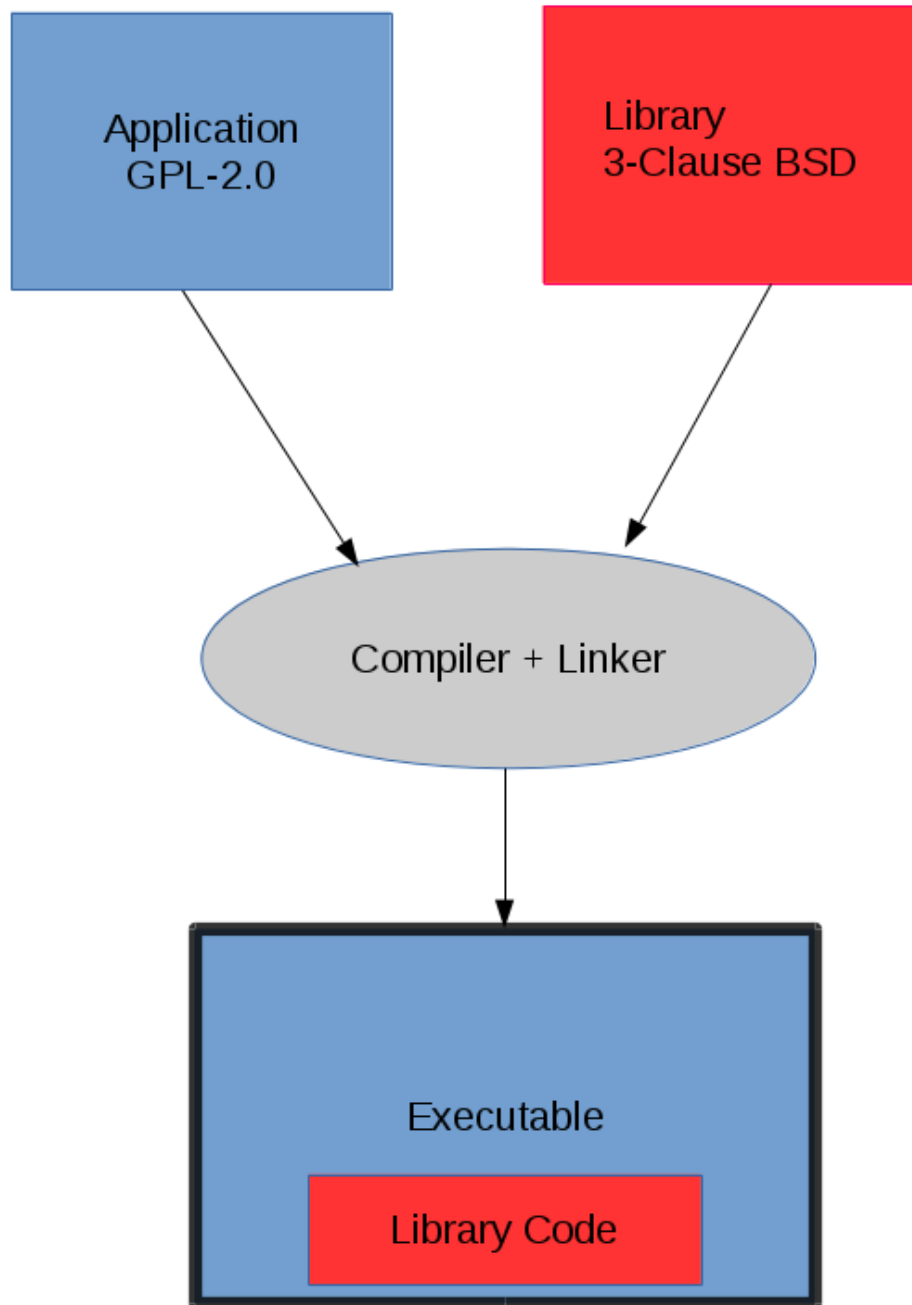
stub_warning (getmsg)
#include <stub-tag.h>
```

# The Impact of Copyleft

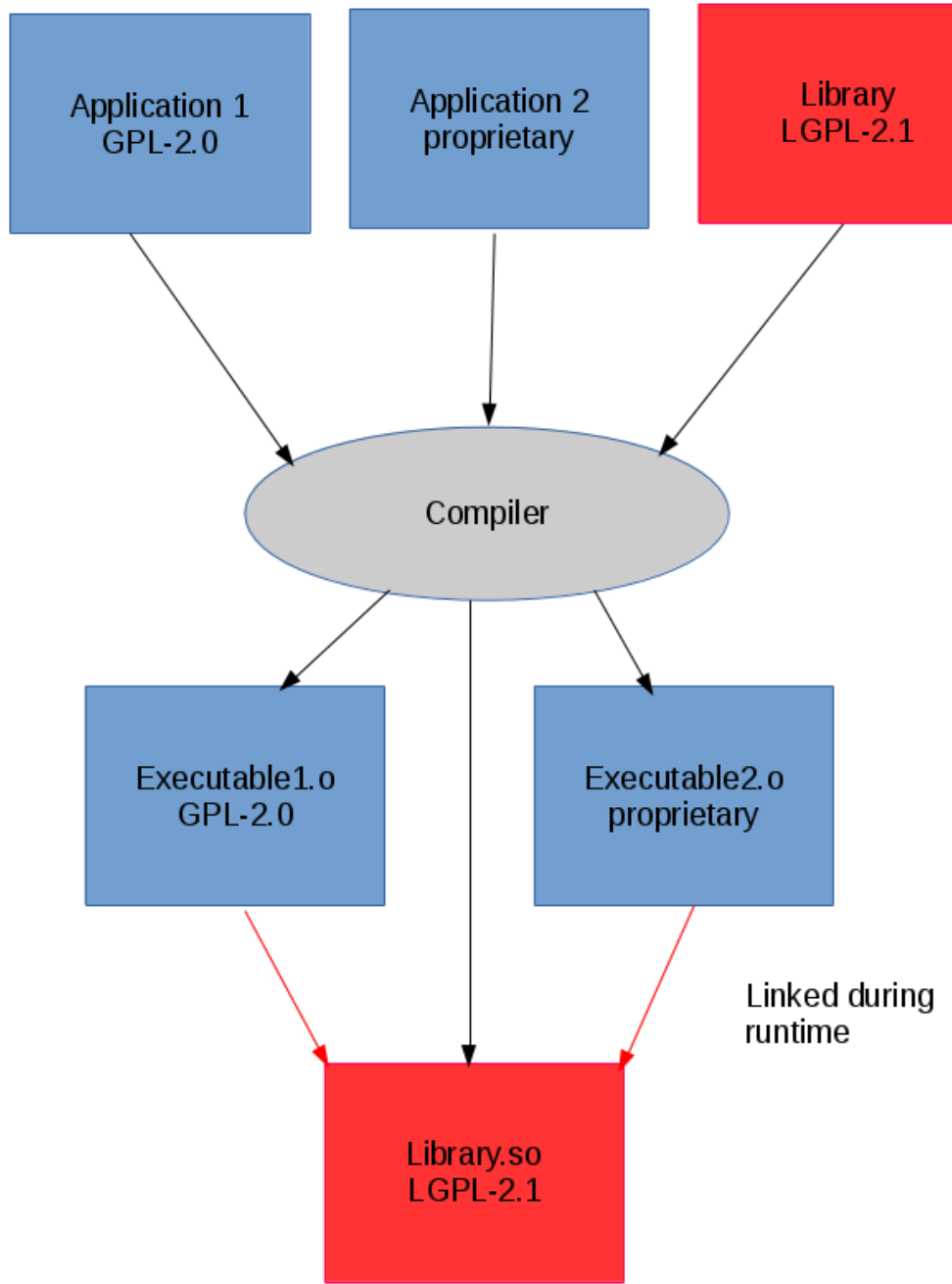
Combination of components:

- Libraries
- Plugins
- Modules

# Static Linking



# Dynamic Linking



# The Impact of Copyleft

Derivative works cannot be licensed under more than one license at the same time (except in the case of dual licensing)

Which licenses are compatible (allowing the use of another license for the modified code)? Which license prevails?

Compliance process: a) are two components to be considered a derivative work (as defined in the respective licenses)?, b) if yes: are the FOSS licenses compatible?



# Non-Copyleft and Non-Copyleft

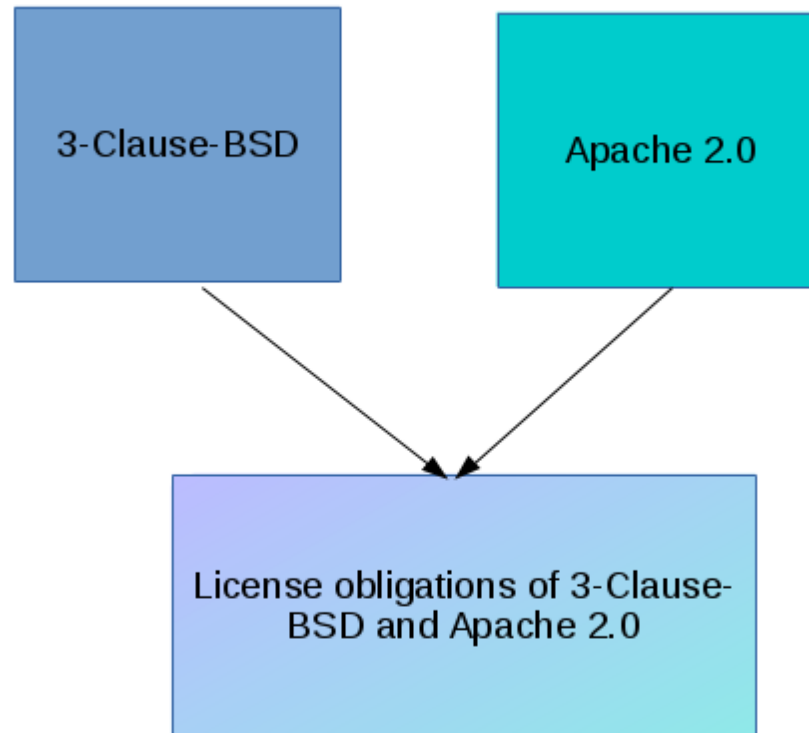
Mixing code under two or more Non-Copyleft licenses

License obligations of all licenses have to be fulfilled altogether

Non-Copyleft licenses are always compatible among each other

Choosing a proprietary license or a Non-Copyleft license for the derivative work

# Non-Copyleft and Non-Copyleft



# Copyleft and Copyleft

Mixing code under two or more copyleft licenses

Code licensed under two different copyleft licenses cannot be combined in one program, as one of the licenses would necessarily have to be violated

Copyleft licenses are not compatible except...

# Copyleft and Copyleft

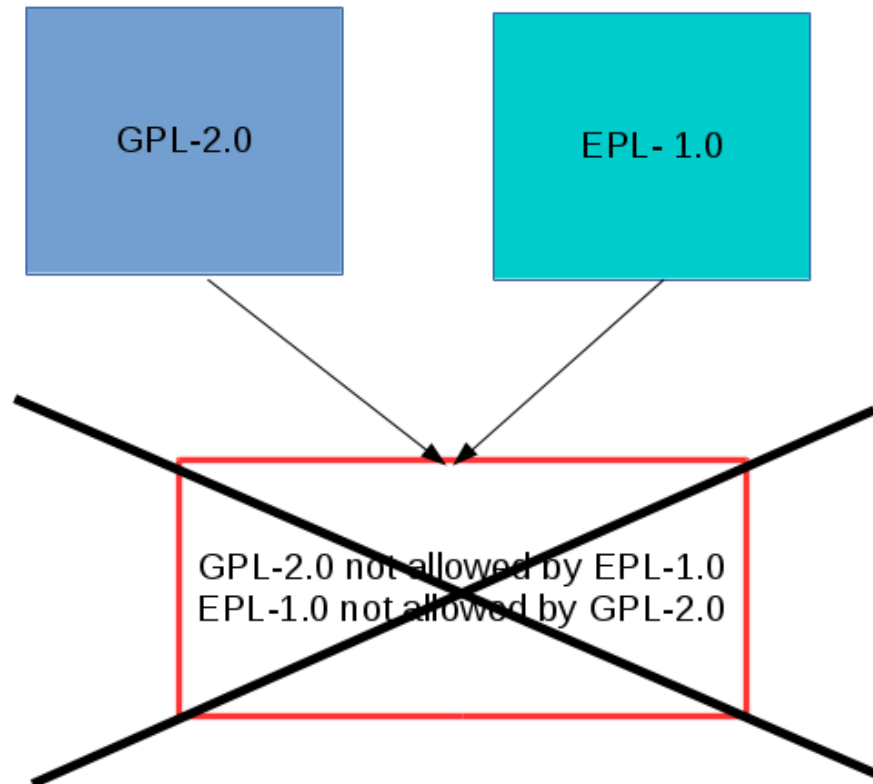
... a specific compatibility clause is allowing the combination

E.g. Sec. 3 LGPL-2.1: *„You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License.“*

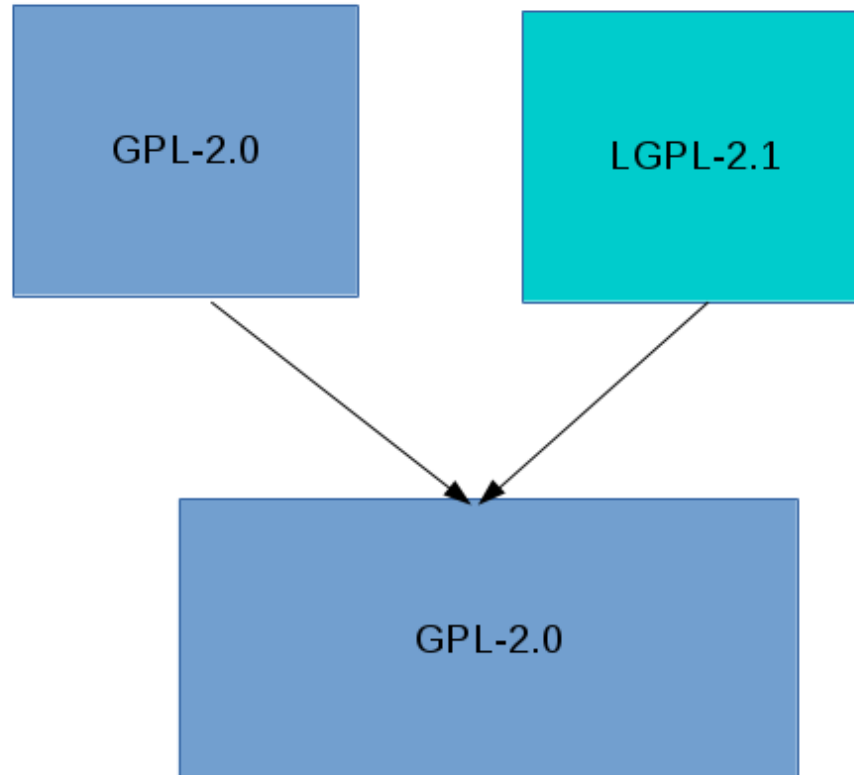
## Copyleft and Copyleft

*Sec. 5 EUPL: „Compatibility clause: If the Licensee Distributes and/or Communicates Derivative Works or copies thereof based upon both the Original Work and another work licensed under a Compatible Licence, this Distribution and/or Communication can be done under the terms of this Compatible Licence. For the sake of this clause, “Compatible Licence” refers to the licences listed in the appendix attached to this Licence. Should the Licensee’s obligations under the Compatible Licence conflict with his/her obligations under this Licence, the obligations of the Compatible Licence shall prevail.“*

# Copyleft and Copyleft



# Copyleft and Copyleft



# Copyleft and Non-Copyleft

Mixing code under a Copyleft license with code under a Non-Copyleft license

Non-Copyleft and Copyleft licenses are compatible but in the case that the Non-Copyleft license provides an obligation the Copyleft license does not provide

Copyleft licenses do not allow additional restrictions (e.g. Sec. 6 GPL-2.0: *„You may not impose any further restrictions on the recipients' exercise of the rights granted herein.“*)



# Copyleft and Non-Copyleft

Example – 4-Clause-BSD license:

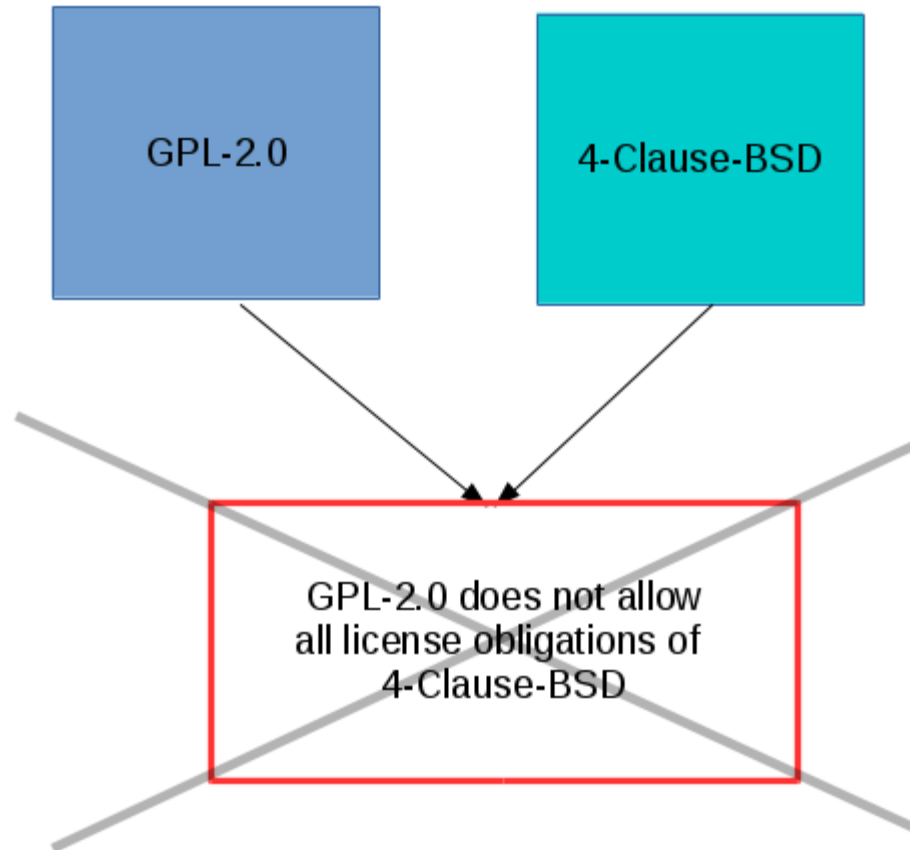
*All advertising materials mentioning features or use of this software must display the following acknowledgement:*

*“This product includes software developed by the the organization.”*

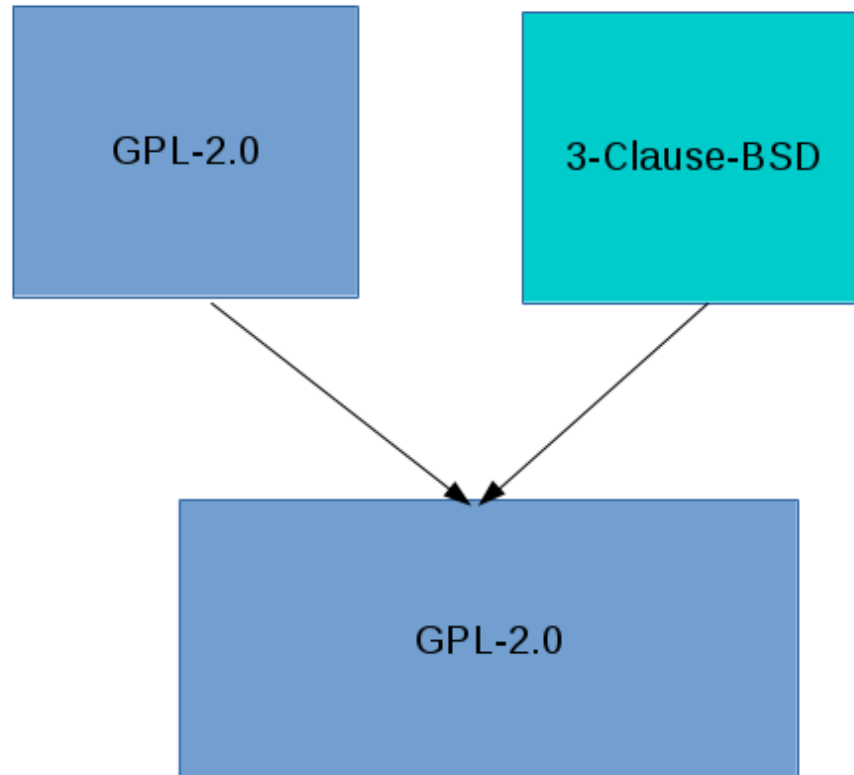
GPL-2.0 does not provide for such an obligation

GPL-2.0 and 4-Clause-BSD are not compatible

# Copyleft and Non-Copyleft



# Copyleft and Non-Copyleft



# License Compatibility

Examination:

- Is there a Copyleft license? Is the Copyleft applicable? (e.g., derivative work) - compatibility of the licenses involved
- More than one Copyleft license?
- If yes: compatibility clause applicable?
- If not: contain(s) the Non-Copyleft license(s) restrictions not contained in the Copyleft license

# License Compatibility

GPL-2.0 (Linux) and GPL-3.0 – internal compatibility

Sec. 9 GPL-2.0 provides as follows:

- "any later version" – license under the GPL-3.0 possible
- Program does not specify a version number – license under all versions (GPL-1.0, GPL-2.0 and GPL-3.0) possible
- GPL-2.0 only? FSF tolerates this licensing for a long time – GPL-2.0 and GPL-3.0 are not compatible in this case

# License Compatibility

New solution for the compatibility problem in the GPL-3.0

## Sec. 7 GPL-3.0

*"Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:*

*a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or*

*b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or ...."*

# License Compatibility

	GPL-2.0	GPL-3.0	LGPL-2.1	LGPL-3.0	Apache License 2.0	MIT License and BSD License (3 clause)	MPL-2.0 <sup>1</sup>	CPL/EPL
GPL-2.0		-	+	-	-	+	+	-
GPL-3.0	-		+	+	+	+	+	-
LGPL-2.1	+	+		+	-	+	+	-
LGPL-3.0	-	+	+		+	+	+	-
Apache License 2.0	-	+	-	+		+	-	-
MIT License and BSD License (3 clause)	+	+	+	+	+		+	+
MPL-2.0 <sup>1</sup>	+	+	+	+	-	+		-
CPL/EPL	-	-	-	-	-	+	-	

<sup>1</sup> Depending on Exhibit B

# Thank you for your attention!

JBB Rechtsanwälte

Dr. Till Jaeger

Christinenstraße 18/19

10119 Berlin

[www.jbb.de](http://www.jbb.de)