

the revolution will be
forked

scott chacon



a short history of
open source

let's say you're a
lawyer

Preferred Stock Investment Agreement

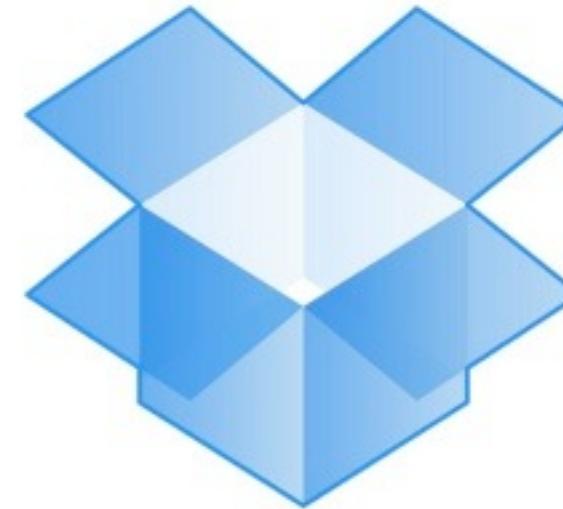
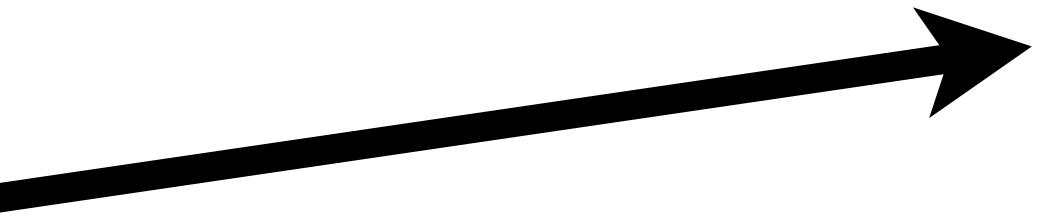
The parties hereto hereby agree as follows:

1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion

Preferred Stock Investment Agreement

The parties hereto hereby agree as follows:

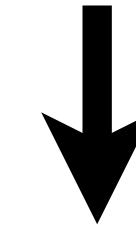
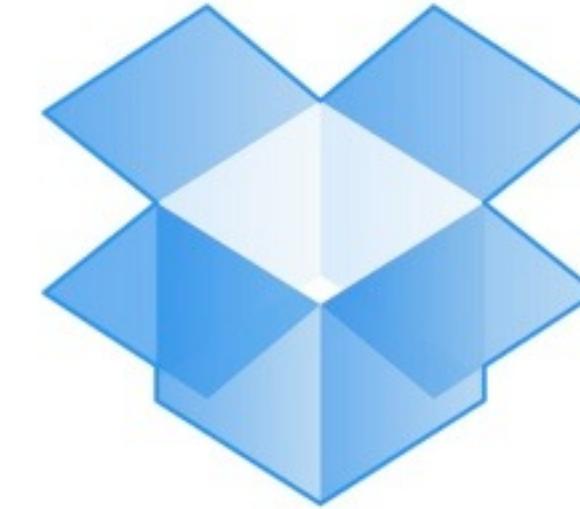
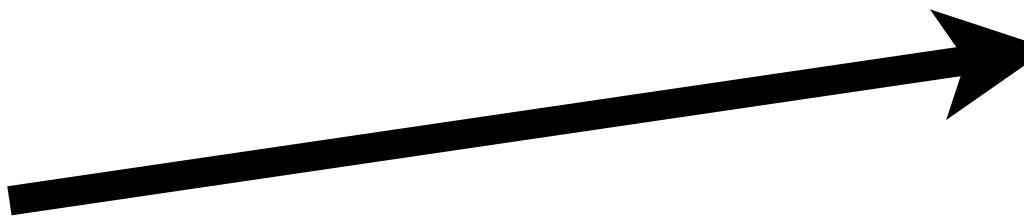
1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion



Preferred Sock Investment Agreement

The parties hereto hereby agree as follows:

1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion



Preferred Sock Investment Agreement

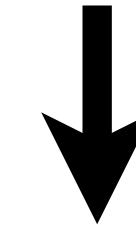
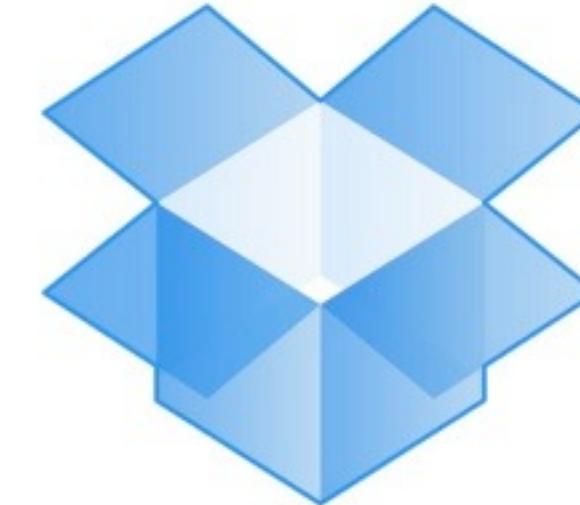
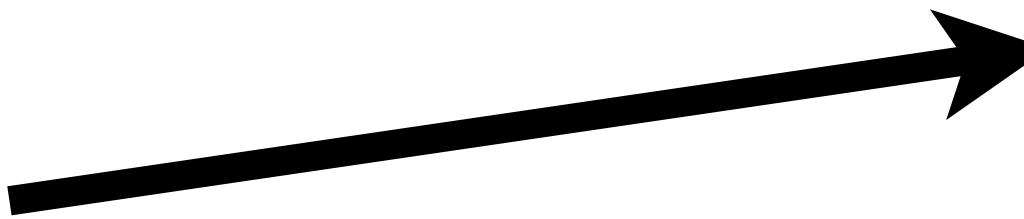
The parties hereto hereby agree as follows:

1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion

Preferred Sock Investment Agreement

The parties hereto hereby agree as follows:

1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion



Preferred Stock Investment Agreement

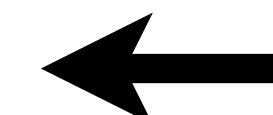
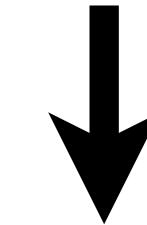
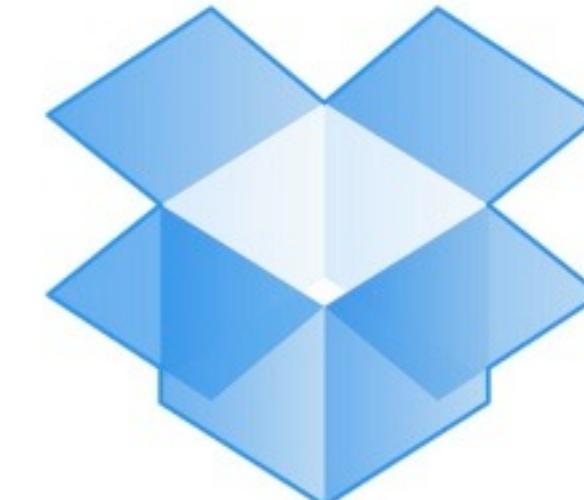
The parties hereto hereby agree as follows:

1. **DEFINITIONS.** Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. **INVESTMENT.** Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion

Preferred Sock Investment Agreement

The parties hereto hereby agree as follows:

1. DEFINITIONS. Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. INVESTMENT. Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion



Preferred Stock Investment Agreement

The parties hereto hereby agree as follows:

1. DEFINITIONS. Capitalized terms used and not otherwise defined in this Agreement or the Exhibit and Schedules thereto shall have the meanings set forth in Exhibit A hereto.
2. INVESTMENT. Subject to the terms and conditions of this Agreement, including the Agreement Terms set forth in Exhibit B hereto, (i) each Purchaser agrees to purchase at the applicable Closing and the Company agrees to sell and issue to each Purchaser at such Closing that number of shares of Series Seed Preferred Stock set forth opposite such Purchaser's name on Schedule 1, at a price per share equal to the Purchase Price (subject to any discounts applicable where all or a portion

a-patchy
way

a-patchy
way
1985 - now

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

bob's laptop

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies-2.txt

```
4.5 cups flour
2   tsp baking soda
3   cups butter
2.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

bob's laptop

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies-2.txt

```
4.5 cups flour
2   tsp baking soda
3   cups butter
2.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

bob's laptop

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies-2.txt

```
4.5 cups flour
2   tsp baking soda
3   cups butter
2.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

bob's laptop

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies-2.txt

```
4.5 cups flour
2   tsp baking soda
3   cups butter
2.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```

diff

aptop

freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies.txt

```
4.5 cups flour
2   tsp baking soda
2   cups butter
1.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



freedom-cookies-2.txt

```
4.5 cups flour
2   tsp baking soda
3   cups butter
2.5 cups brown sugar
0.5 cup white sugar
4   eggs
2   tsp vanilla extract
4   cups chocolate chips
```



better-cookies.patch

```
@@ -1,7 +1,7 @@
 2   tsp baking soda
-2   cups butter
-1.5 cups brown sugar
+3   cups butter
+2.5 cups brown sugar
 0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

patch

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

bob's laptop

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```



freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies.txt

```
4.5 cups flour
2 tsp baking soda
2 cups butter
1.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

freedom-cookies-2.txt

```
4.5 cups flour
2 tsp baking soda
3 cups butter
2.5 cups brown sugar
0.5 cup white sugar
4 eggs
2 tsp vanilla extract
4 cups chocolate chips
```

better-cookies.patch

```
@@ -1,7 +1,7 @@
2 tsp baking soda
-2 cups butter
-1.5 cups brown sugar
+3 cups butter
+2.5 cups brown sugar
0.5 cup white sugar
```

contribution model

inconsistent

error prone

complex

How to Get Your Change Into the Linux Kernel
or
Care And Operation Of Your Linus Torvalds

For a person or company who wishes to submit a change to the Linux kernel, the process can sometimes be daunting if you're not familiar with "the system." This text is a collection of suggestions which can greatly increase the chances of your change being accepted.

Read Documentation/SubmitChecklist for a list of items to check before submitting code. If you are submitting a driver, also read Documentation/SubmittingDrivers.

SECTION 1 - CREATING AND SENDING YOUR CHANGE

1) "diff -up"

Use "diff -up" or "diff -uprn" to create patches.

All changes to the Linux kernel occur in the form of patches, as generated by diff(1). When creating your patch, make sure to create it in "unified diff" format, as supplied by the '-u' argument to diff(1). Also, please use the '-p' argument which shows which C function each change is in - that makes the resultant diff a lot easier to read. Patches should be based in the root kernel source directory, not in any lower subdirectory.

To create a patch for a single file, it is often sufficient to do:

```
SRCTREE= linux-2.6
MYFILE= drivers/net/mydriver.c
cd $SRCTREE
cp $MYFILE $MYFILE.orig
vi $MYFILE # make your change
cd ..
diff -up $SRCTREE/$MYFILE.{orig,} > /tmp/patch
```

To create a patch for multiple files, you should unpack a "vanilla", or unmodified kernel source tree, and generate a diff against your own source tree. For example:

```
MYSRC= /devel/linux-2.6
tar xvzf linux-2.6.12.tar.gz
mv linux-2.6.12 linux-2.6.12-vanilla
diff -upN -X linux-2.6.12-vanilla/Documentation/dontdiff \
linux-2.6.12-vanilla $MISRC > /tmp/patch
```

"dontdiff" is a list of files which are generated by the kernel during the build process, and should be ignored in any diff(1)-generated patch. The "dontdiff" file is included in the kernel tree in 2.6.12 and later.

Make sure your patch does not include any extra files which do not belong in a patch submission. Make sure to review your patch -after- generated it with diff(1), to ensure accuracy.

If your changes produce a lot of deltas, you may want to look into splitting them into individual patches which modify things in logical stages. This will facilitate easier reviewing by other kernel developers, very important if you want your patch accepted. There are a number of scripts which can aid in this:

Quilt:
<http://savannah.nongnu.org/projects/quilt>

Andrew Morton's patch scripts:
<http://userweb.kernel.org/~akpm/stuff/patch-scripts.tar.gz>
Instead of these scripts, quilt is the recommended patch management tool (see above).

2) Describe your changes.

Describe the technical detail of the change(s) your patch includes.

Be as specific as possible. The WORST descriptions possible include things like "update driver X", "bug fix for driver X", or "this patch includes updates for subsystem X. Please apply."

The maintainer will thank you if you write your patch description in a form which can be easily pulled into Linux's source code management system, git, as a "commit log". See #15, below.

If your description starts to get long, that's a sign that you probably need to split up your patch. See #3, next.

When you submit or resubmit a patch or patch series, include the complete patch description and justification for it. Don't just say that this is version N of the patch (series). Don't expect the patch merger to refer back to earlier patch versions or referenced URLs to find the patch description and put that into the patch. I.e., the patch (series) and its description should be self-contained. This benefits both the patch merger(s) and reviewers. Some reviewers probably didn't even receive earlier versions of the patch.

If the patch fixes a logged bug entry, refer to that bug entry by number and URL.

3) Separate your changes.

Separate _logical changes_ into a single patch file.

For example, if your changes include both bug fixes and performance enhancements for a single driver, separate those changes into two or more patches. If your changes include an API update, and a new

On the other hand, if you make a single change to numerous files, group those changes into a single patch. Thus a single logical change is contained within a single patch.

If one patch depends on another patch in order for a change to be complete, that is OK. Simply note "this patch depends on patch X" in your patch description.

If you cannot condense your patch set into a smaller set of patches, then only post say 15 or so at a time and wait for review and integration.

4) Style check your changes.

Check your patch for basic style violations, details of which can be found in Documentation/CodingStyle. Failure to do so simply wastes the reviewers time and will get your patch rejected, probably without even being read.

At a minimum you should check your patches with the patch style checker prior to submission (scripts/checkpatch.pl). You should be able to justify all violations that remain in your patch.

5) Select e-mail destination.

Look through the MAINTAINERS file and the source code, and determine if your change applies to a specific subsystem of the kernel, with an assigned maintainer. If so, e-mail that person. The script scripts/get_maintainer.pl can be very useful at this step.

If no maintainer is listed, or the maintainer does not respond, send your patch to the primary Linux kernel developer's mailing list, linux-kernel@vger.kernel.org. Most kernel developers monitor this e-mail list, and can comment on your changes.

Do not send more than 15 patches at once to the vger mailing lists!!!

Linus Torvalds is the final arbiter of all changes accepted into the Linux kernel. His e-mail address is <torvalds@linux-foundation.org>. He gets a lot of e-mail, so typically you should do your best to -avoid- sending him e-mail.

Patches which are bug fixes, are "obvious" changes, or similarly require little discussion should be sent or CC'd to Linus. Patches which require discussion or do not have a clear advantage should usually be sent first to linux-kernel. Only after the patch is discussed should the patch then be submitted to Linus.

6) Select your CC (e-mail carbon copy) list.

Unless you have a reason NOT to do so, CC linux-kernel@vger.kernel.org.

Other kernel developers besides Linus need to be aware of your change, so that they may comment on it and offer code review and suggestions. linux-kernel is the primary Linux kernel developer mailing list.

Other mailing lists are available for specific subsystems, such as USB, framebuffer devices, the VFS, the SCSI subsystem, etc. See the MAINTAINERS file for a mailing list that relates specifically to your change.

Majordomo lists of VGER.KERNEL.ORG at:
<http://vger.kernel.org/vger-lists.html>

If changes affect userland-kernel interfaces, please send the MAN-PAGES maintainer (as listed in the MAINTAINERS file) a man-pages patch, or at least a notification of the change, so that some information makes its way into the manual pages.

Even if the maintainer did not respond in step #5, make sure to ALWAYS copy the maintainer when you change their code.

For small patches you may want to CC the Trivial Patch Monkey trivial@kernel.org which collects "trivial" patches. Have a look into the MAINTAINERS file for its current manager.

Trivial patches must qualify for one of the following rules:

Spelling fixes in documentation
Spelling fixes which could break grep(1)
Warning fixes (cluttering with useless warnings is bad)
Compilation fixes (only if they are actually correct)
Runtime fixes (only if they actually fix things)
Removing use of deprecated functions/macros (eg. check_region)
Contact detail and documentation fixes
Non-portable code replaced by portable code (even in arch-specific, since people copy, as long as it's trivial)
Any fix by the author/maintainer of the file (ie. patch monkey in re-transmission mode)

then you just add a line saying

Signed-off-by: Random J Developer <random@developer.example.org>
using your real name (sorry, no pseudonyms or anonymous contributions.)

A Reviewed-by tag is a statement of opinion that the patch is an appropriate modification of the kernel without any remaining serious technical issues. Any interested reviewer (who has done the work) can offer a Reviewed-by tag for a patch. This tag serves to give credit to reviewers and to inform maintainers of the degree of review which has been done on the patch. Reviewed-by tags, when supplied by reviewers known to understand the subject area and to perform thorough reviews, will normally increase the likelihood of your patch getting into the kernel.

If you are a subsystem or branch maintainer, sometimes you need to slightly modify patches you receive in order to merge them, because the code is not exactly the same in your tree and the submitters'. If you stick strictly to rule (c), you should ask the submitter to rediff, but this is a totally counter-productive waste of time and energy. Rule (b) allows you to adjust the code, but then it is very impolite to change one submitter's code and make him endorse your bugs. To solve this problem, it is recommended that you add a line between the last Signed-off-by header and yours, indicating the nature of your changes. While there is nothing mandatory about this, it seems like prepending the description with your mail and/or name, all enclosed in square brackets, is noticeable enough to make it obvious that

you are responsible for last-minute changes. Example :

See Documentation/email-clients.txt for hints about configuring your e-mail client so that it sends your patches untouched.

8) E-mail size.

8) E-mail size.

When sending patches to Linus, always follow step #7.

Large changes are not appropriate for mailing lists, and some maintainers. If your patch, uncompressed, exceeds 300 kB in size, it is preferred that you store your patch on an Internet-accessible server, and provide instead a URL (link) pointing to your patch.

9) Name your kernel version.

It is important to note, either in the subject line or in the patch description, the kernel version to which this patch applies.

If the patch does not apply cleanly to the latest kernel version, Linus will not apply it.

10) Don't get discouraged. Re-submit.

After you have submitted your change, be patient and wait. If Linus likes your change and applies it, it will appear in the next version of the kernel that he releases.

However, if your change doesn't appear in the next version of the kernel, there could be any number of reasons. It's YOUR job to narrow down those reasons, correct what was wrong, and submit your updated change.

It is quite common for Linus to "drop" your patch without comment. That's the nature of the system. If he drops your patch, it could be due to

- * Your patch did not apply cleanly to the latest kernel version.
- * Your patch was not sufficiently discussed on linux-kernel.
- * A style issue (see section 2).
- * An e-mail formatting issue (re-read this section).
- * A technical problem with your change.
- * He gets tons of e-mail, and yours got lost in the shuffle.
- * You are being annoying.

When in doubt, solicit comments on linux-kernel mailing list.

11) Include PATCH in the subject

Due to high e-mail traffic to Linus, and to linux-kernel, it is common convention to prefix your subject line with [PATCH]. This lets Linus and other kernel developers more easily distinguish patches from other e-mail discussions.

12) Sign your work

To improve tracking of who did what, especially with patches that can percolate to their final resting place in the kernel through several layers of maintainers, we've introduced a "sign-off" procedure on patches that are being emailed around.

The sign-off is a simple line at the end of the explanation for the patch, which certifies that you wrote it or otherwise have the right to pass it on as an open-source patch. The rules are pretty simple: if you can certify the below:

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

The explanation body will be committed to the permanent source changelog, so should make sense to a competent reader who has long since forgotten the immediate details of the discussion that might have led to this patch. Including symptoms of the failure which the patch addresses (kernel log messages, oops messages, etc.) is especially useful for people who might be searching the commit logs looking for the applicable patch. If a patch fixes a compile failure, it may not be necessary to include _all_ of the compile failures; just enough that it is likely that someone searching for the patch can find it. As in the "summary phrase", it is important to be both succinct as well as descriptive.

The "---" marker line serves the essential purpose of marking for patch handling tools where the changelog message ends.

One good use for the additional comments after the "---" marker is for a diffstat, to show what files have changed, and the number of inserted and deleted lines per file. A diffstat is especially useful on bigger patches. Other comments relevant only to the moment or the maintainer, not suitable for the permanent changelog, should also go here. A good example of such comments might be "patch changelog" which describe what has changed between the v1 and v2 version of the patch.

If you are going to include a diffstat after the "---" marker, please use diffstat options "-p 1 -w 70" so that filenames are listed from the top of the kernel source tree and don't use too much horizontal space (easily fit in 80 columns, maybe with some indentation).

A Suggested-by tag indicates that the patch idea is suggested by the person named and ensures credit to the person for the idea. Please note that this tag should not be added without the reporter's permission, especially if the idea was not posted in a public forum. That said, if we diligently credit our idea reporters, they will, hopefully, be inspired to help us again in the future.

See more details on the proper patch format in the following references.

16) Sending "git pull" requests (from Linus emails)

Please write the git repo address and branch name alone on the same line

15) The canonical patch format

The canonical patch subject line is:

Subject: [PATCH 001/123] subsystem: summary phrase

The canonical patch message body contains the following:

- A "from" line specifying the patch author.
- An empty line.
- The body of the explanation, which will be copied to the permanent changelog to describe this patch.
- The "Signed-off-by:" lines, described above, which will also go in the changelog.
- A marker line containing simply "---".
- Any additional comments not suitable for the changelog.
- The actual patch (diff output).

The Subject line format makes it very easy to sort the emails alphabetically by subject line - pretty much any email reader will support that - since the sequence number is zero-padded, the numerical and alphabetic sort is the same.

13) When to use Acked-by: and Cc:

The Signed-off-by: tag indicates that the signer was involved in the development of the patch, or that he/she was in the patch's delivery path.

If a person was not directly involved in the preparation or handling of a patch but wishes to signify and record their approval of it then they can arrange to have an Acked-by: line added to the patch's changelog.

Acked-by: is often used by the maintainer of the affected code when that maintainer neither contributed to nor forwarded the patch.

Acked-by: is not as formal as Signed-off-by:. It is a record that the acker has at least reviewed the patch and has indicated acceptance. Hence patch mergers will sometimes manually convert an acker's "yep, looks good to me" into an Acked-by:.

For these reasons, the "summary" must be no more than 70-75 characters, and it must describe both what the patch changes, as well as why the patch might be necessary. It is challenging to be both succinct and descriptive, but that is what a well-written summary should do.

If a person has had the opportunity to comment on a patch, but has not provided such comments, you may optionally add a "Cc:" tag to the patch. This is the only tag which might be added without an explicit action by the person it names. This tag documents that potentially interested parties have been included in the discussion

14) Using Reported-by:, Tested-by:, Reviewed-by: and Suggested-by:

If this patch fixes a problem reported by somebody else, consider adding a Reported-by: tag to credit the reporter for their contribution. Please note that this tag should not be added without the reporter's permission, especially if the problem was not reported in a public forum. That said, if we diligently credit our bug reporters, they will, hopefully, be inspired to help us again in the future.

A couple of example Subjects:

Subject: [patch 2/5] ext2: improve scalability of bitmap searching

Subject: [PATCHv2 001/207] x86: fix eflags tracking

The "from" line must be the very first line in the message body, and has the form:

From: Original Author <author@example.com>

The "from" line specifies who will be credited as the author of the patch in the permanent changelog. If the "from" line is missing, then the "From:" line from the email header will be used to determine the patch author in the changelog.

The explanation body will be committed to the permanent source changelog, so should make sense to a competent reader who has long since forgotten the immediate details of the discussion that might have led to this patch. Including symptoms of the failure which the patch addresses (kernel log messages, oops messages, etc.) is especially useful for people who might be searching the commit logs looking for the applicable patch. If a patch fixes a compile failure, it may not be necessary to include _all_ of the compile failures; just enough that it is likely that someone searching for the patch can find it. As in the "summary phrase", it is important to be both succinct as well as descriptive.

The "---" marker line serves the essential purpose of marking for patch

How to Get Your Change Into the Linux Kernel
or
Care And Operation Of Your Linus Torvalds

For a person or company who wishes to submit a change to the Linux kernel, the process can sometimes be daunting if you're not familiar with "the system." This text is a collection of suggestions which can greatly increase the chances of your change being accepted.

Read Documentation/SubmitChecklist for a list of items to check before submitting code. If you are submitting a driver, also read Documentation/SubmittingDrivers.

SECTION 1 - CREATING AND SENDING YOUR CHANGE

1) "diff -up"

Use "diff -up" or "diff -uprn" to create patches.

All changes to the Linux kernel occur in the form of patches, as generated by diff(1). When creating your patch, make sure to create it in "unified diff" format, as supplied by the '-u' argument to diff(1). Also, please use the '-p' argument which shows which C function each change is in - that makes the resultant diff a lot easier to read.

To create a patch for a single file, this is often sufficient to do:

```
SRCTREE=linux-2.6
MYFILE=drivers/net/mydriver.c
cp $MYFILE $MYFILE.orig
vi $MYFILE # make your change
cd ..
diff -up $SRCTREE/$MYFILE.{orig,} > /tmp/patch
```

To create a patch for multiple files, you should unpack a "vanilla", or unmodified kernel source tree, and generate a diff against your own source tree. For example:

```
MYSRC=/devel/linux-2.6
xvfz linux-2.6.12.tar.gz
cd linux-2.6.12
patch -X 1 -p1 -i vger-mailinglist/dontdiff.patch
cp -uN $MYFILE $MYFILE.orig
cp $MYFILE $MYFILE.C > /tmp/patch
```

"dontdiff" is a list of files which are generated by the kernel during the build process, and should be ignored in any diff(1)-generated patch. The "-f" flag is needed in the kernel since 2.6.12 and later.

Make sure your patch does not include any extra files which do not belong in a patch submission. Make sure to review your patch -after- generated it with diff(1), to ensure accuracy.

If your changes produce a lot of deltas, you may want to look into splitting them into individual patches which modify things in logical stages. This will facilitate easier reviewing by other kernel developers, very important if you want your patch accepted. There are a number of scripts which can aid in this:

Quilt:
<http://savannah.nongnu.org/projects/quilt>

Andrew Morton's patch scripts:
<http://userweb.kernel.org/~akpm/stuff/patch-scripts.tar.gz>
Instead of these scripts, quilt is the recommended patch management tool (see above).

2) Describe your changes.

Describe the technical detail of the change(s) your patch includes.

Be as specific as possible. The WORST descriptions possible include things like "update driver X", "bug fix for driver X", or "this patch includes updates for subsystem X. Please apply."

The maintainer will thank you if you write your patch description in a form which can be easily pulled into Linux's source code management system, git, as a "commit log". See #15, below.

If your description starts to get long, that's a sign that you probably need to split up your patch. See #3, next.

When you submit or resubmit a patch or patch series, include the complete patch description and justification for it. Don't just say that this is version N of the patch (series). Don't expect the patch merger to refer back to earlier patch versions or referenced URLs to find the patch description and put that into the patch. I.e., the patch (series) and its description should be self-contained. This benefits both the patch merger(s) and reviewers. Some reviewers probably didn't even receive earlier versions of the patch.

If the patch fixes a logged bug entry, refer to that bug entry by number and URL.

3) Separate your changes.

Separate _logical_ changes_ into a single patch file.

For example, if your changes include both bug fixes and performance enhancements for a single driver, separate those changes into two or more patches. If your changes include an API update, and a new

On the other hand, if you make a single change to numerous files, group those changes into a single patch. Thus a single logical change is contained within a single patch.

If one patch depends on another patch in order for a change to be complete, that is OK. Simply note "this patch depends on patch X" in your patch description.

If you cannot condense your patch set into a smaller set of patches, then only post say 15 or so at a time and wait for review and integration.

4) Style check your changes.

Check your patch for basic style violations, details of which can be found in Documentation/CodingStyle. Failure to do so simply wastes the reviewers time and will get your patch rejected, probably without even being read.

At a minimum you should check your patches with the patch style checker prior to submission (scripts/checkpatch.pl). You should be able to justify all violations that remain in your patch.

5) Select e-mail destination.

Look through the MAINTAINERS file and the source code, and determine if your change applies to a specific subsystem of the kernel, with an assigned maintainer. If so, e-mail that person. The script scripts/get_maintainer.pl can be very useful at this step.

If no maintainer is listed, or the maintainer does not respond, send your patch to the primary Linux kernel developer's mailing list, linux-kernel@vger.kernel.org. Most kernel developers monitor this e-mail list, and respond to submissions on various mailing lists.

Do not send more than 15 patches at once to the vger mailing lists!

Linus Torvalds is a final arbiter of changes accepted into the Linux kernel. His e-mail address is <torvalds@linux-foundation.org>. He gets a lot of e-mail, so typically you should do your best to avoid sending him e-mail.

Patches which are bug fixes, are "obvious" changes, or similarly require little discussion should be sent or CC'd to Linus. Patches which require discussion or do not have a clear advantage should usually be sent first to linux-kernel. Only after the patch is discussed should the patch then be submitted to Linus.

11) Include PATCH in the subject

Due to high e-mail traffic to Linus, and to linux-kernel, it is common convention to prefix your subject line with [PATCH]. This lets Linus and other kernel developers more easily distinguish patches from other e-mail discussions.

12) Sign your work

When tracking of who did what, especially with patches that can percolate to their final resting place in the kernel through several layers of maintainers, we've introduced a "sign-off" procedure on what is being emailed around.

The sign-off is a simple line at the end of the explanation for the patch, which certifies that you wrote it or otherwise have the right to pass it on as an open-source patch. The rules are pretty simple: if you can certify the below:

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

Even if the maintainer did not respond in step #5, make sure to ALWAYS copy the maintainer when you change their code.

For small patches you may want to CC the Trivial Patch Monkey trivial@kernel.org which collects "trivial" patches. Have a look into the MAINTAINERS file for its current manager.

Trivial patches must qualify for one of the following rules:

Spelling fixes in documentation

Spelling fixes which could break grep(1)

Warning fixes (cluttering with useless warnings is bad)

Compilation fixes (only if they are actually correct)

Runtime fixes (only if they actually fix things)

Removing use of deprecated functions/macros (eg. check_region)

Contact detail and documentation fixes

Non-portable code replaced by portable code (even in arch-specific, since people copy, as long as it's trivial)

Any fix by the author/maintainer of the file (ie. patch monkey in re-transmission mode)

then you just add a line saying

Signed-off-by: Random J Developer <random@developer.example.org>

using your real name (sorry, no pseudonyms or anonymous contributions.)

A Reviewed-by tag is a statement of opinion that the patch is an appropriate modification of the kernel without any remaining serious technical issues. Any interested reviewer (who has done the work) can offer a Reviewed-by tag for a patch. This tag serves to give credit to reviewers and to inform maintainers of the degree of review which has been done on the patch. Reviewed-by tags, when supplied by reviewers known to understand the subject area and to perform thorough reviews, will normally increase the likelihood of your patch getting into the kernel.

If you are a subsystem or branch maintainer, sometimes you need to slightly modify patches you receive in order to merge them, because the code is not exactly the same in your tree and the submitters'. If you stick strictly to rule (c), you should ask the submitter to rediff, but this is a totally counter-productive waste of time and energy. Rule (b) allows you to adjust the code, but then it is very impolite to change one submitter's code and make him endorse your bugs. To solve this problem, it is recommended that you add a line between the last Signed-off-by header and yours, indicating the nature of your changes. While there is nothing mandatory about this, it seems like prepending the description with your mail and/or name, all enclosed in square brackets, is noticeable enough to make it obvious that you are responsible for last-minute changes. Example :

Signed-off-by: Random J Developer <random@developer.example.org> [lucky@maintainer.example.org; struct foo moved from foo.c to foo.h]

Signed-off-by: Lucky K Maintainer <lucky@maintainer.example.org>

8) E-mail size.

When sending patches to Linus, always follow step #7.

Large changes are not appropriate for mailing lists, and some maintainers. If your patch, uncompressed, exceeds 300 kB in size, it is preferred that you store your patch on an Internet-accessible server, and provide instead a URL (link) pointing to your patch.

9) Name your kernel version.

It is important to note, either in the subject line or in the patch description, the kernel version to which this patch applies.

If the patch does not apply cleanly to the latest kernel version, Linus will not apply it.

10) Don't get discouraged. Re-submit.

After you have submitted your change, be patient and wait. If Linus likes your change and applies it, it will appear in the next version of the kernel that he releases.

However, if your change doesn't appear in the next version of the kernel, there could be any number of reasons. It's YOUR job to narrow down those reasons, correct what was wrong, and submit your updated change.

It is quite common for Linus to "drop" your patch without comment. That's the nature of the system. If he drops your patch, it could be

* Your patch did not apply cleanly to the latest kernel version.

* Your patch was accidentally applied to the linux-kernel.

* A style issue (see section 2).

* An e-mail for getting issued (re-read this section).

* An e-mail for getting merged (re-read this section).

* He gets tons of e-mail, and yours got lost in the shuffle.

* You are being annoying.

When in doubt, solicit comments on linux-kernel mailing list.

11) Include PATCH in the subject

Due to high e-mail traffic to Linus, and to linux-kernel, it is common convention to prefix your subject line with [PATCH]. This lets Linus and other kernel developers more easily distinguish patches from other e-mail discussions.

12) Sign your work

When tracking of who did what, especially with patches that can percolate to their final resting place in the kernel through several layers of maintainers, we've introduced a "sign-off" procedure on what is being emailed around.

The sign-off is a simple line at the end of the explanation for the patch, which certifies that you wrote it or otherwise have the right to pass it on as an open-source patch. The rules are pretty simple: if you can certify the below:

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

Even if the maintainer did not respond in step #5, make sure to ALWAYS copy the maintainer when you change their code.

For small patches you may want to CC the Trivial Patch Monkey trivial@kernel.org which collects "trivial" patches. Have a look into the MAINTAINERS file for its current manager.

Trivial patches must qualify for one of the following rules:

Spelling fixes in documentation

Spelling fixes which could break grep(1)

Warning fixes (cluttering with useless warnings is bad)

Compilation fixes (only if they are actually correct)

Runtime fixes (only if they actually fix things)

Removing use of deprecated functions/macros (eg. check_region)

Contact detail and documentation fixes

Non-portable code replaced by portable code (even in arch-specific, since people copy, as long as it's trivial)

Any fix by the author/maintainer of the file (ie. patch monkey in re-transmission mode)

then you just add a line saying

Signed-off-by: Random J Developer <random@developer.example.org>

using your real name (sorry, no pseudonyms or anonymous contributions.)

A Reviewed-by tag is a statement of opinion that the patch is an appropriate modification of the kernel without any remaining serious technical issues. Any interested reviewer (who has done the work) can offer a Reviewed-by tag for a patch. This tag serves to give credit to reviewers and to inform maintainers of the degree of review which has been done on the patch. Reviewed-by tags, when supplied by reviewers known to understand the subject area and to perform thorough reviews, will normally increase the likelihood of your patch getting into the kernel.

If you are a subsystem or branch maintainer, sometimes you need to slightly modify patches you receive in order to merge them, because the code is not exactly the same in your tree and the submitters'. If you stick strictly to rule (c), you should ask the submitter to rediff, but this is a totally counter-productive waste of time and energy. Rule (b) allows you to adjust the code, but then it is very impolite to change one submitter's code and make him endorse your bugs. To solve this problem, it is recommended that you add a line between the last Signed-off-by header and yours, indicating the nature of your changes. While there is nothing mandatory about this, it seems like prepending the description with your mail and/or name, all enclosed in square brackets, is noticeable enough to make it obvious that you are responsible for last-minute changes. Example :

Signed-off-by: Random J Developer <random@developer.example.org> [lucky@maintainer.example.org; struct foo moved from foo.c to foo.h]

Signed-off-by: Lucky K Maintainer <lucky@maintainer.example.org>

15) The canonical patch format

The canonical patch subject line is:

Subject: [PATCH 001/123] subsystem: summary phrase

The canonical patch message body contains the following:

- A "from" line specifying the patch author.
- An empty line.
- The body of the explanation, which will be copied to the permanent changelog to describe this patch.
- The "Signed-off-by:" lines, described above, which will also go in the changelog.
- A marker line containing simply "---".
- Any additional comments not suitable for the changelog.
- The actual patch (diff output).

The Subject line format makes it very easy to sort the emails alphabetically by subject line - pretty much any email reader will support that - since the sequence number is zero-padded, the numerical and alphabetic sort is the same.

The "subsystem" in the email's Subject should identify which area or subsystem of the kernel is being patched.

A significant exception is when moving code from another -- in this case you should not modify the same patch which moves it. This clearly delineates the code itself.

Check your patches with the patch style checker pri-

arily using identifier for the patch. It propagates the way

globally unique identifier for the patch. It propagates the way

into the global changelog. The "summary phrase" may be used in developer discussions which refer to the patch. People will want to google for the "summary phrase" to read discussion regarding that

patch. It will be the one thing people will be looking for when that

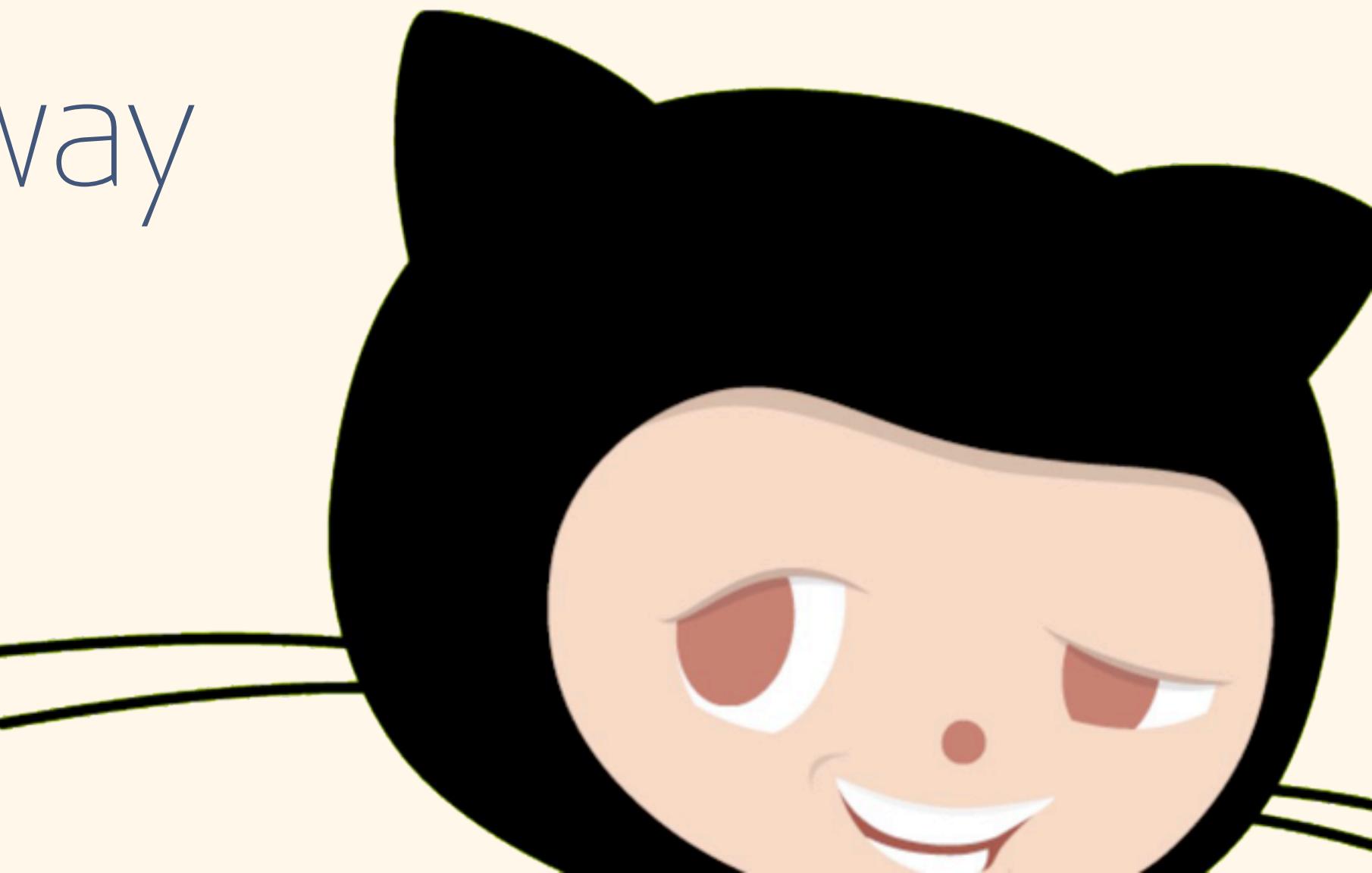
patch is merged for the first time. It is a good idea to have a brief description of the patch and its purpose. When

two or three months later, they are going through perhaps

thousands of patches using tools such as "gitk" or "git log

how **open source** has
changed

the
GitHubway



openness

openness
collaboration

Author
Publishes

WhiteHouse/petitions · GitHub

GitHub, Inc. [US] https://github.com/WhiteHouse/petitions

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC WhiteHouse / petitions Watch Star 702 Fork 123

Code Network Pull Requests 3 Issues 17 Graphs

Drupal installation profile powering We The People at petitions.whitehouse.gov — Read more
<https://petitions.whitehouse.gov/>

Clone in Mac ZIP HTTP SSH Git Read-Only git://github.com/WhiteHouse/petitions.git Read-Only access

branch: 7.x-2.x Files Commits Branches 4 Tags 60

petitions / + 216 commits

PT-805 - Updated the users resource to retrieve count metadata inform... ...
theneonlobster authored 21 days ago latest commit ae187641a7

modules 21 days ago PT-805 - Updated the users resource to retrieve count metadata inform... [theneonlobster]

selenium 10 months ago Add petition install profile files. [WH-NewMedia]

themes a month ago Updated CSS for API Gallery. [bryanhirsch]

INSTALL.md a month ago Updated installation profile to 7.x-2.0-beta8 [theneonlobster]

README.md a month ago Updated installation profile to 7.x-2.0-beta8 [theneonlobster]

UPDATE.txt 21 days ago PT-805 - Updated the users resource to retrieve count metadata inform... [theneonlobster]

build-petitions make 21 days ago PT-805 - Updated the users resource to retrieve count metadata inform... [theneonlobster]

Author
Publishes

Fork

Contributor
Modifies

jpmckinney/petitions - GitHub

GitHub, Inc. [US] https://github.com/jpmckinney/petitions

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC jpmckinney / petitions forked from WhiteHouse/petitions Watch Star 0 Fork 123

Code Network Pull Requests 0 Graphs

Drupal installation profile powering We The People at petitions.whitehouse.gov — Read more
<https://petitions.whitehouse.gov/>

Clone in Mac ZIP HTTP SSH Git Read-Only git://github.com/jpmckinney/petitions.git Read-Only access

branch: 7.x-2.x Files Commits Branches 3 Tags 37

petitions / + 109 commits

respect word boundaries in profanity detector

jpmckinney authored 2 months ago latest commit 9bf19fd5c0

| File | Date | Message |
|------------|---------------|--|
| make | 8 months ago | Update make file for 7.x-1.0-alpha22 release. [bryanhirsch] |
| modules | 2 months ago | respect word boundaries in profanity detector [jpmckinney] |
| selenium | 10 months ago | Add petition install profile files. [WH-NewMedia] |
| themes | 5 months ago | modifying htaccess rule to allow fb image to be accessed over http. [theneonlobster] |
| INSTALL.md | 7 months ago | fix minor typos in README file [sheldonrampton] |
| README.md | 6 months ago | Update README.md [welschp] |

jpmckinney/petitions - GitHub

GitHub, Inc. [US] https://github.com/jpmckinney/petitions

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC jpmckinney / petitions forked from WhiteHouse/petitions Watch Star 0 Fork 123

Code Network Pull Requests 0 Graphs

Drupal installation profile powering We The People at petitions.whitehouse.gov — Read more
<https://petitions.whitehouse.gov/>

Clone in Mac ZIP HTTP SSH Git Read-Only git://github.com/jpmckinney/petitions.git Read-Only access

branch: 7.x-2.x Files Commits Branches 3 Tags 37

petitions / + 109 commits

respect word boundaries in profanity detector

jpmckinney authored 2 months ago latest commit 9bf19fd5c0

| File | Date | Description |
|------------|---------------|--|
| make | 8 months ago | Update make file for 7.x-1.0-alpha22 release. [bryanhirsch] |
| modules | 2 months ago | respect word boundaries in profanity detector [jpmckinney] |
| selenium | 10 months ago | Add petition install profile files. [WH-NewMedia] |
| themes | 5 months ago | modifying htaccess rule to allow fb image to be accessed over http. [theneonlobster] |
| INSTALL.md | 7 months ago | fix minor typos in README file [sheldonrampton] |
| README.md | 6 months ago | Update README.md [welschp] |

jpmckinney/petitions · GitHub

GitHub, Inc. [US] https://github.com/jpmckinney/petitions

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC  **jpmckinney / petitions** forked from WhiteHouse/petitions

Watch Star 0 Fork 123

Code Network Pull Requests 0 Graphs

Drupal installation profile powering We The People at petitions.whitehouse.gov — Read more
<https://petitions.whitehouse.gov/>

Clone in Mac ZIP HTTP SSH Git Read-Only git://github.com/jpmckinney/petitions.git Read-Only access

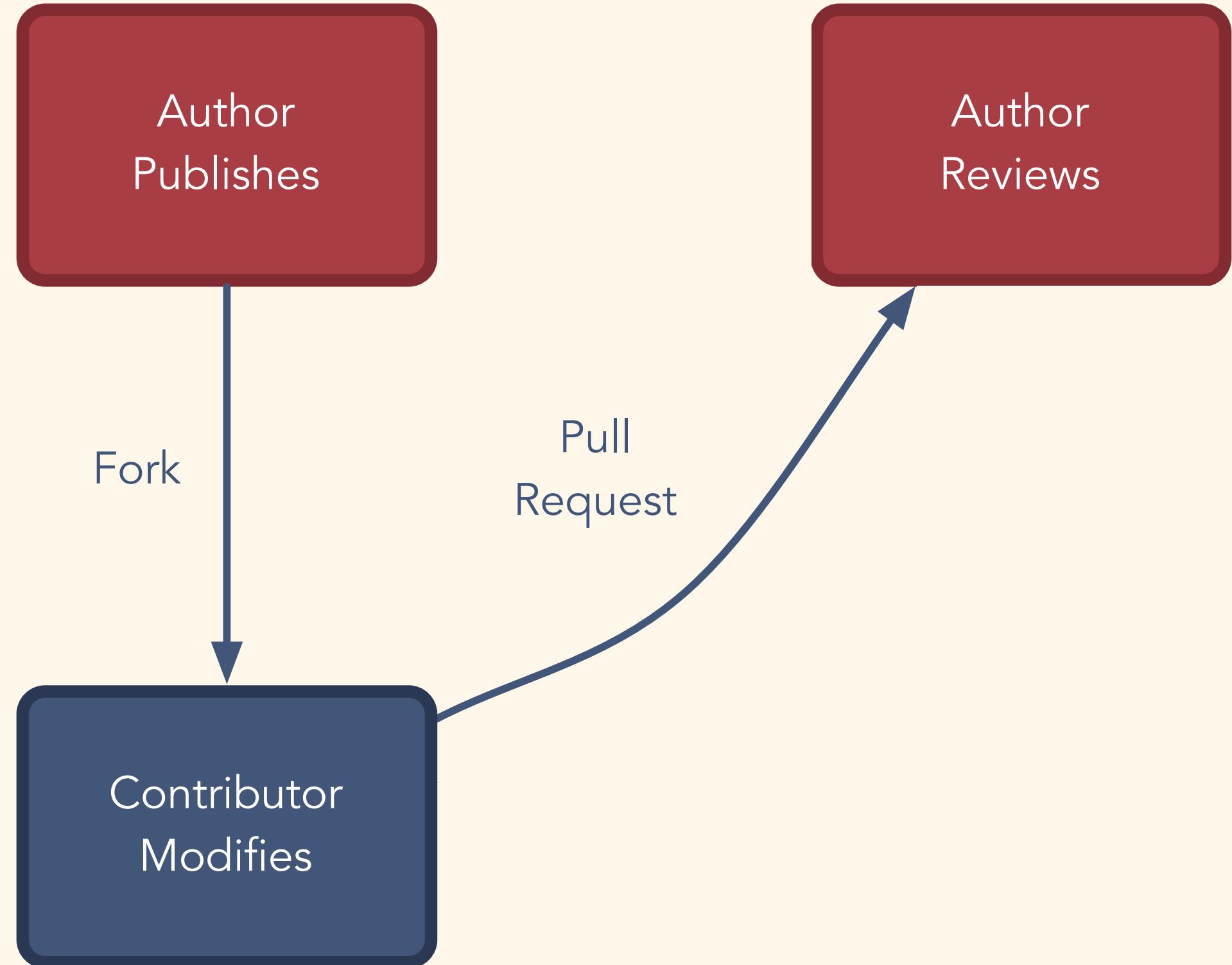
branch: 7.x-2.x Files Commits Branches 3 Tags 37

petitions / + 109 commits

respect word boundaries in profanity detector

 **jpmckinney** authored 2 months ago latest commit 9bf19fd5c0 

| | | |
|--|---------------|--|
|  make | 8 months ago | Update make file for 7.x-1.0-alpha22 release. [bryanhirsch] |
|  modules | 2 months ago | respect word boundaries in profanity detector [jpmckinney] |
|  selenium | 10 months ago | Add petition install profile files. [WH-NewMedia] |
|  themes | 5 months ago | modifying htaccess rule to allow fb image to be accessed over http. [theneonlobster] |
|  INSTALL.md | 7 months ago | fix minor typos in README file [sheldonrampton] |
|  README.md | 6 months ago | Update README.md [welschp] |



Respect word boundaries | X

GitHub, Inc. [US] https://github.com/WhiteHouse/petitions/pull/53

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC WhiteHouse / petitions Watch Star 702 Fork 123

Code Network Pull Requests 3 Issues 17 Graphs

Open jpmckinney wants to merge 1 commit into WhiteHouse:7.x-2.x from jpmckinney:7.x-2.x 2 #53

Discussion - Commits 1 Files Changed 1

jpmckinney opened this pull request 2 months ago

Respect word boundaries in profanity detector

No one is assigned No milestone

See #51. This change will avoid a massive number of false positives. The old code would flag anything with the words "classic", "grass", "passion", "assure" and hundreds of other words, because they all contain the substring "ass". I can provide similar lists of non-profanity words for other words in the default list of profanities.

Open + 1 addition - 1 deletion

2 participants

jpmckinney added a commit 2 months ago

jpmckinney respect word boundaries in profanity detector 9bf19fd

Respect word boundaries X

GitHub, Inc. [US] https://github.com/WhiteHouse/petitions/pull/53/files

This repository Search or type a command Explore Gist Blog Help schacon

PUBLIC WhiteHouse / petitions Watch Star 702 Fork 123

Code Network Pull Requests 3 Issues 17 Graphs

Open jpmckinney wants to merge 1 commit into WhiteHouse:7.x-2.x from jpmckinney:7.x-2.x 2 #53

Discussion Commits 1 Files Changed 1

Showing 1 changed file with 1 addition and 1 deletion. Show Diff Stats

modules/custom/is_profanity/is_profanity.module

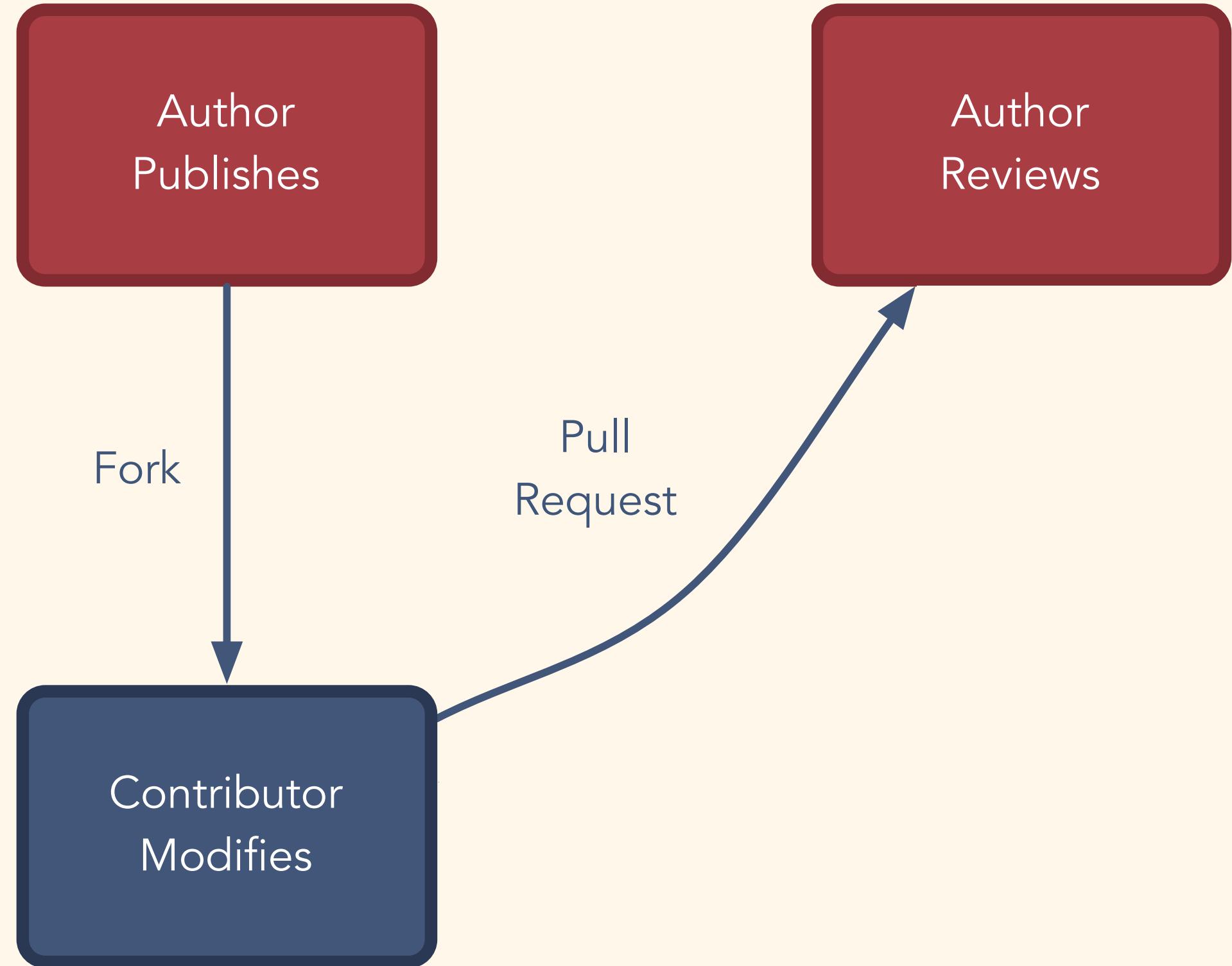
View file @ 9bf19fd

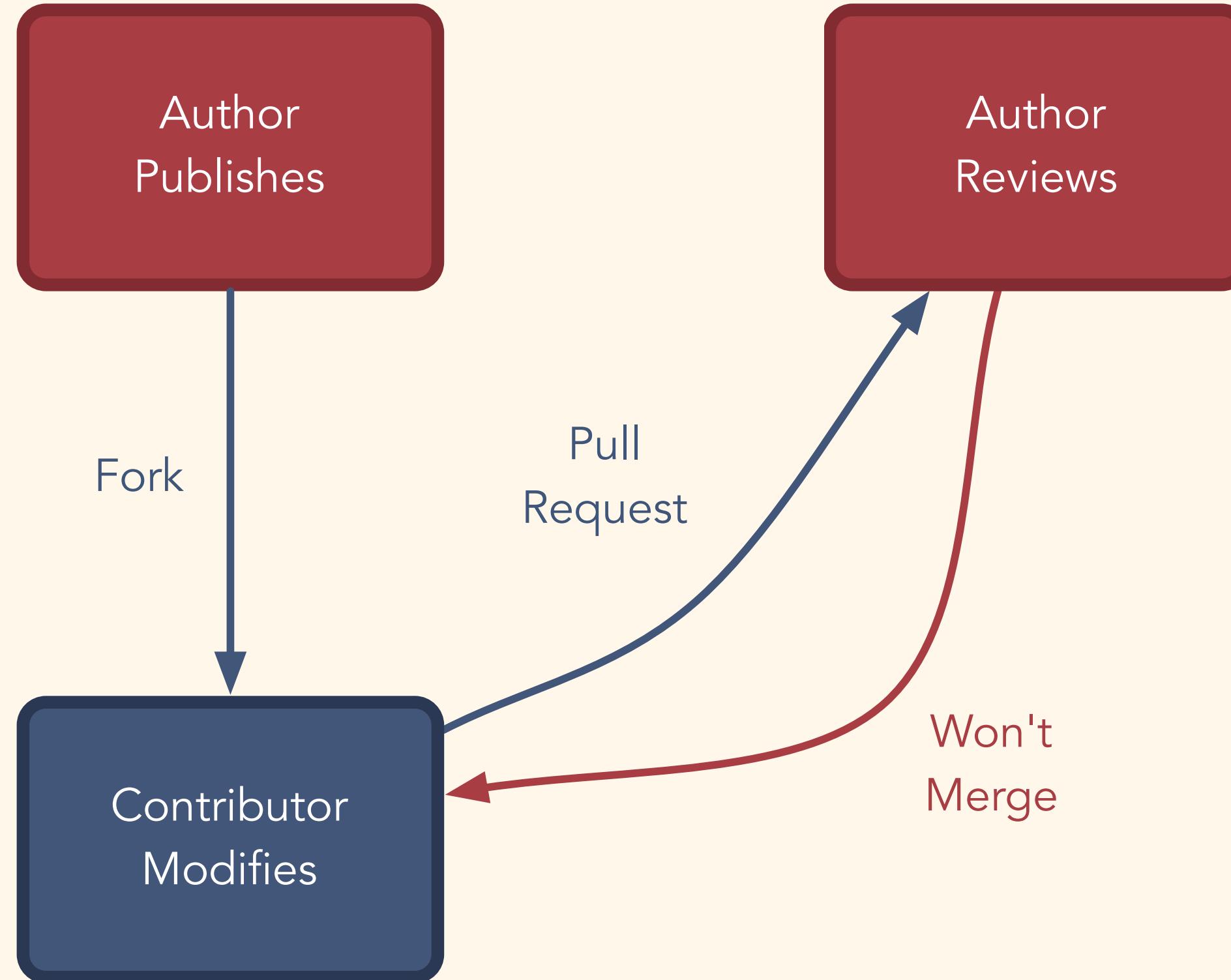
| | | |
|---|---|---|
| 2 | 1 | 2 |
|---|---|---|

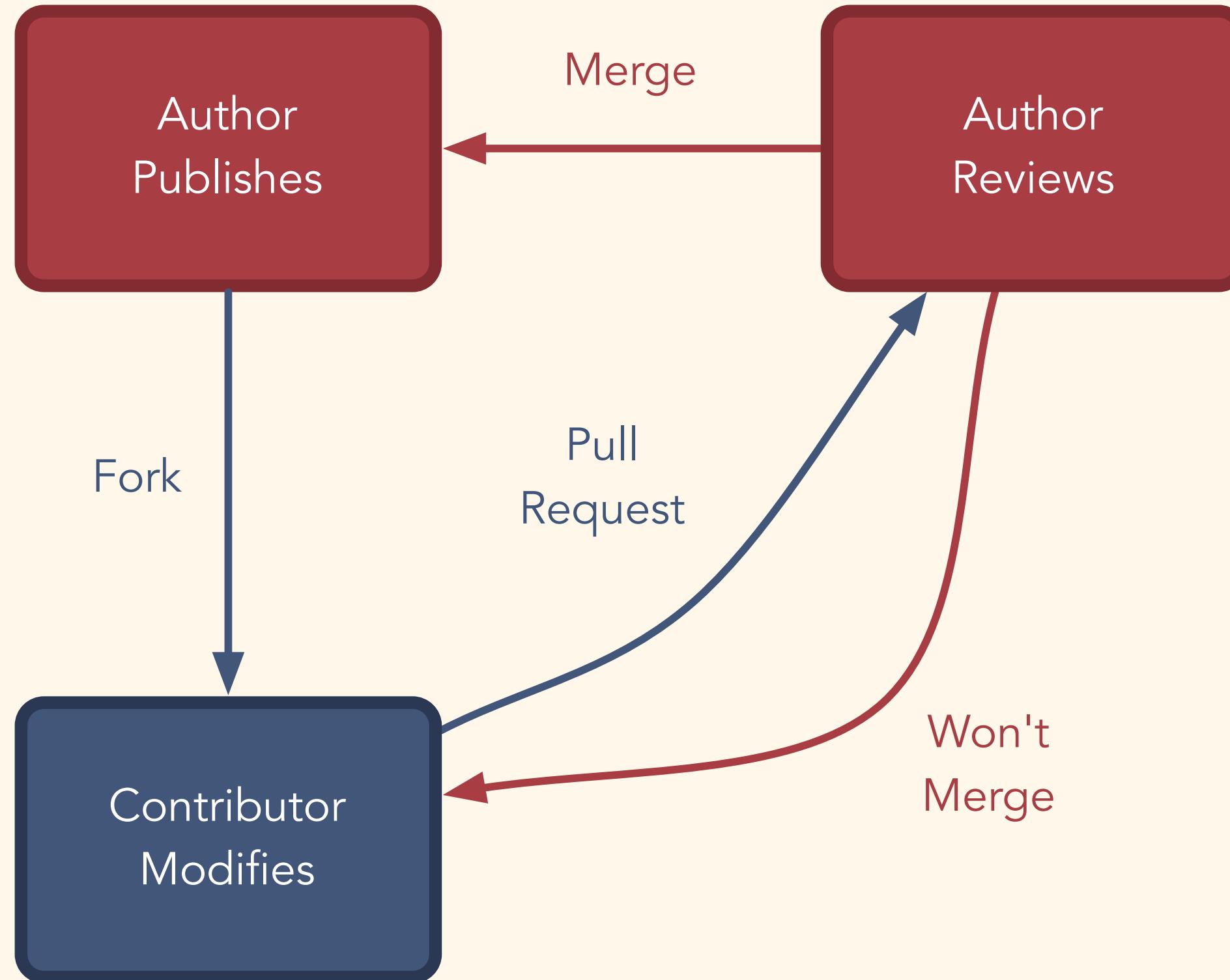
```
@@ -60,7 +60,7 @@ function is_profanity($string, $terms = array()) {  
 60 60 }  
 61 61 foreach ($terms as $term) {  
 62 62   $term = str_replace(',', '', $term);  
 63 - if (strpos($string, $term) !== FALSE) {  
 63 + if (preg_match("/\b$string\b/", $term)) {  
 64 64     return TRUE;  
 65 65   }  
 66 66 }
```

+ 38 58
+ 39 59
+ 40 60

Tip: You can add notes to lines in a file.
Hover to the left of a line to make a note







contribution model

simple

open

standardized

WhiteHouse/petitions/README.md

Contributing

Anyone is encouraged to contribute to the project by [forking](#) and submitting a pull request. (If you are new to GitHub, you might start with a [basic tutorial](#).)

By contributing to this project, you grant a world-wide, royalty-free, perpetual, irrevocable, non-exclusive, transferable license to all users under the terms of the [Gnu General Public License v2 or later](#).

All comments, messages, pull requests, and other submissions received through official White House pages including this GitHub page are subject to the Presidential Records Act and may be archived. Learn more <http://WhiteHouse.gov/privacy>

WhiteHouse/petitions/README.md

Contributing

Anyone is encouraged to contribute to the project by [forking](#) and submitting a pull request. (If you are new to GitHub, you might start with a [basic tutorial](#).)

By contributing to this project, you grant a world-wide, royalty-free, perpetual, irrevocable, non-exclusive, transferable license to all users under the terms of the [Gnu General Public License v2 or later](#).

All comments, messages, pull requests, and other submissions received through official White House pages including this GitHub page are subject to the Presidential Records Act and may be archived. Learn more <http://WhiteHouse.gov/privacy>

More people contribute to more projects

More people contribute to more projects

More about the contribution,
less about the process

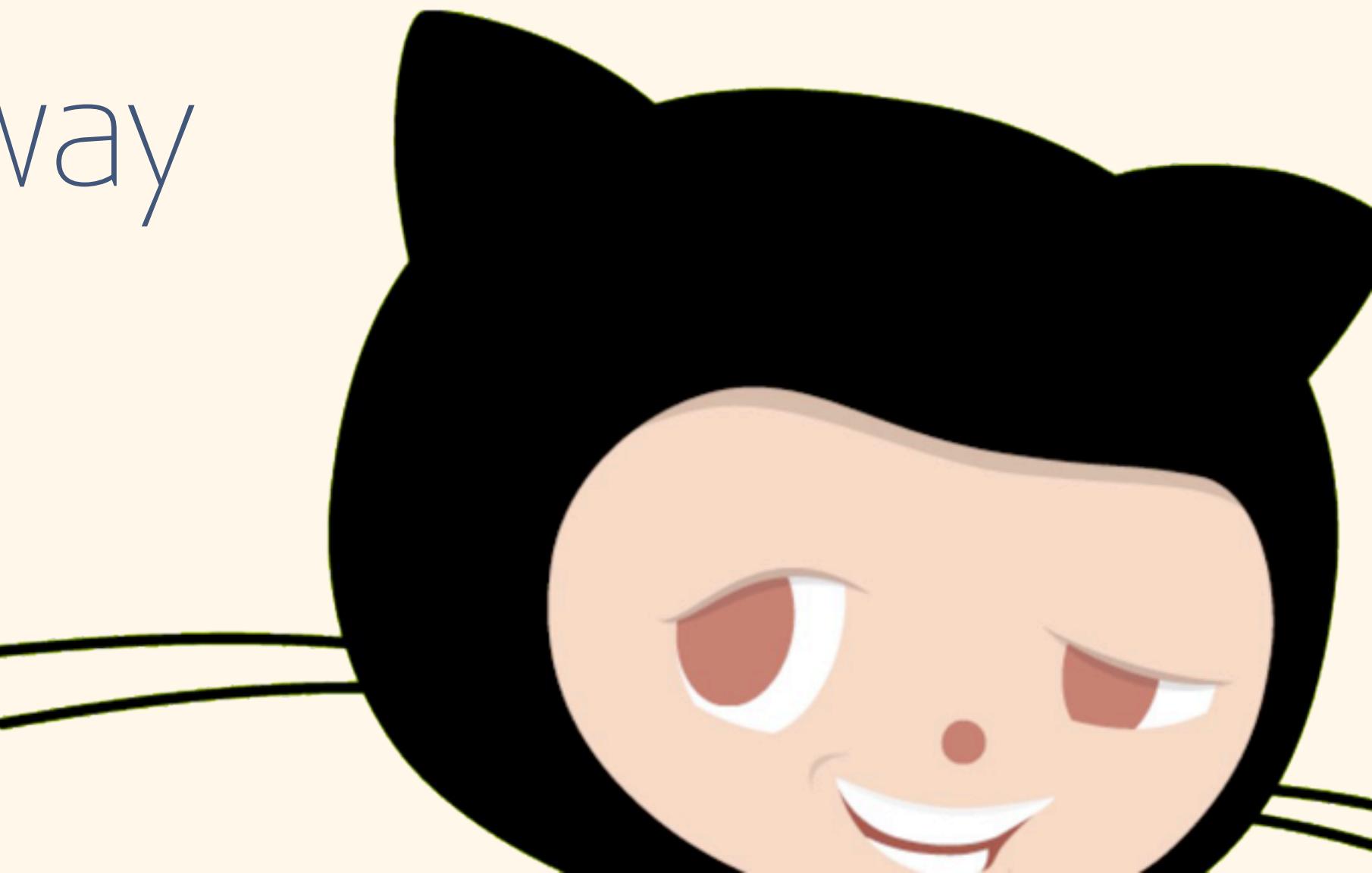
More people contribute to more projects

More about the contribution,
less about the process

Clear path to implementation

how **government** will
change

the
GitHubway



the
GitHubway
(government style)



openness

openness

collaboration

white house open data policy





EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
WASHINGTON, D.C. 20503

THE DIRECTOR

May 9, 2013

M-13-13

MEMORANDUM FOR THE HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES

FROM:

Sylvia M. Burwell
Director

Steven VanRoekel
Federal Chief Information Officer

Todd Park
U.S. Chief Technology Officer

Dominic J. Mancini
Acting Administrator, Office of Information and Regulatory Affairs

SUBJECT: Open Data Policy—Managing Information as an Asset

Information is a valuable national resource and a strategic asset to the Federal Government, its partners, and the public. In order to ensure that the Federal Government is taking full advantage of its information resources, executive departments and agencies (hereafter referred to as “agencies”) must manage information as an asset throughout its life cycle to promote openness and interoperability, and properly safeguard systems and information. Managing government information as an asset will increase operational efficiencies, reduce costs, improve services, support mission needs, safeguard personal information, and increase public access to valuable government information.

Making information resources accessible, discoverable, and usable by the public can help fuel entrepreneurship, innovation, and scientific discovery – all of which improve Americans’ lives and contribute significantly to job creation. For example, decades ago, the Federal Government made both weather data and the Global Positioning System (GPS) freely available to anyone. Since then, American entrepreneurs and innovators have used these resources to create navigation systems, weather newscasts and warning systems, location-based applications, precision farming tools, and much more.

Pursuant to Executive Order of May 9, 2013, *Making Open and Machine Readable the New Default for Government Information*, this Memorandum establishes a framework to help institutionalize the principles of effective information management at each stage of the information’s life cycle to promote interoperability and openness. Whether or not particular information can be made public, agencies can apply this framework to all information resources to promote efficiency and produce value.

Project Open Data

project-open-data.github.io

Project Open Data Help Improve this Content View Revision History FAQ

PROJECT OPEN DATA

Open Data Policy – Managing Information as an Asset

1. Background

Data is a valuable national resource and a strategic asset to the U.S. Government, its partners, and the public. Managing this data as an asset and making it available, discoverable, and usable – [in a word, open](#) – not only strengthens our democracy and promotes efficiency and effectiveness in government, but also has the potential to create economic opportunity and improve citizens' quality of life.

For example, when the U.S. Government released weather and GPS data to the public, it fueled an industry that today is valued at tens of billions of dollars per year. Now, weather and mapping tools are ubiquitous and help everyday Americans [navigate their lives](#).

The ultimate value of data can often not be predicted. That's why the U.S. Government released a [policy](#) that instructs agencies to manage their data, and information more generally, as an asset from the start and, wherever possible, release it to the public in a way that makes it open, discoverable, and usable.

The White House developed Project Open Data – this collection of code, tools, and case studies – to help agencies adopt the Open Data Policy and unlock the potential of government data. Project Open Data will evolve over time as a community resource to facilitate broader adoption of open data practices in government. Anyone – government employees, contractors, developers, the general public – can view and contribute. So dive right in and [help to build a better world through the power of open data](#).

2. Definitions

This section is a list of definitions and principles used to guide the project.

2-1 [Open Data Principles](#) - The set of open data principles.

2-2 [Standards, Specifications, and Formats](#) - Standards, specifications, and formats supporting open data objectives.

Editing /index.md at master

prose.io/#project-open-data/project-open-data.github.io/edit/master/index.md

M4 project-open-data / project-open-data.github.io
index.md

h1 **h2** Published ✓

##1. Background

Data is a valuable national resource and a strategic asset to the U.S. Government, its partners, and the public. Managing this data as an asset and making it available, discoverable, and usable – [\[in a word, open!\]\(principles/\)](#) – not only strengthens our democracy and promotes efficiency and effectiveness in government, but also has the potential to create economic opportunity and improve citizens' quality of life.

For example, when the U.S. Government released weather and GPS data to the public, it fueled an industry that today is valued at tens of billions of dollars per year. Now, weather and mapping tools are ubiquitous and help everyday Americans [\[navigate their lives!\]\(business-case/\)](#).

The ultimate value of data can often not be predicted. That's why the U.S. Government released a [\[policy!\]\(policy-memo/\)](#) that instructs agencies to manage their data, and information more generally, as an asset from the start and, wherever possible, release it to the public in a way that makes it open, discoverable, and usable.

The White House developed Project Open Data – this collection of code, tools, and case studies – to help agencies adopt the Open Data Policy and unlock the potential of government data. Project Open Data will evolve over time as a community resource to facilitate broader adoption of open data practices in government. Anyone – government employees, contractors, developers, the general public – can view and contribute. So dive right in and [\[help to build a better world through the power of open data!\]\(governance/\)](#).

##2. Definitions

This section is a list of definitions and principles used to guide the project.

2-1 [\[Open Data Principles!\]\(principles/\)](#) – The set of open data principles.

2-2 [\[Standards, Specifications, and Formats!\]\(open-standards/\)](#) – Standards, specifications, and formats supporting open data objectives.

living, collaborative
document

22

open pull requests

31

closed pull requests

case study #1

adding a link

This repository Search or type a command Explore Gist Blog Help

project-open-data / project-open-data.github.io

Merged haleyvandyck merged 5 commits into project-open-data:master from jnatividad:LinkedDataCook 15 days ago #21

Discussion Commits Files Changed

Added W3C Cookbook for Open Government Linked Data

No one is assigned No milestone

The W3C Cookbook was drafted by the W3C Government Linked Data Working Group

19 participants and others

jnatividad added some commits 25 days ago

- Added that CKAN supports DCAT out-of-the-box 5694b65
- Added W3C Cookbook for Open Government Linked Data a34f6e8
- Revert "Added W3C Cookbook for Open Government Linked Data" 81ddc06
- Added W3C Cookbook for Open Government Linked Data 8085078

benbalter commented 25 days ago

@philipashlock @MarinaMartin @kachok... linked data isn't my thing. I'm all for clean, human readable URLs, but thoughts on adding?

konklone commented 25 days ago

I don't wish to start a holy war here, but Linked Data isn't at the level I'd like to see as an officially endorsed recommendation for government agencies. The scope of this project, of this Order, is to get data published in open, machine-readable formats where the barrier to use the data is as low as possible. Keeping recommendations at the transport level (JSON, XML, CSV, not binary, etc.) makes a lot more sense than the schema level (FOAF, RDF, OWL, etc.). It leaves more flexibility in both publishing and consumption, and focuses interoperability efforts on more fundamental concerns (like unique identifiers).

kachok commented 25 days ago

I completely agree with Eric. The only place where anything resembling linked data will make sense at this moment is to restrict datasets IDs to be URLs. Will help tremendously in harvesting and metadata reconciliation between diff data.json files.

sbms44 commented 25 days ago

Agreed with Dmitry and Eric. This project shouldn't push any standards that haven't won broad acceptance among the types of engineers in the target audience. Linked Data clearly doesn't yet qualify. There's a real danger of inhibiting use with overly baroque standards – the lackluster use of the SEC's XBRL disclosures outside of the enterprise software world should be a sobering example.

URLs enforce a lot of good and necessary identifier habits, though, so I'm on board for that as well.

georgethomas commented 25 days ago

Guys, agencies are already doing LD, with more cutting through the unintentional FUD created by developers (that typically have never done any LD work, or otherwise have a limited understanding of what it's all about) and bringing more LD based on voluntary consensus open standard machine readable (and interpretable) formats every day :-)

The structure of resource representations / messages / document objects, regardless of whether they're serialized as XML, JSON, CSV, etc are about schema, not transport. The lack of machine readable metadata and data structure is a huge complaint from open gov data consumers that LD fixes without breaking a sweat. Take a look at the core metadata page -

From my world view, the items above outline the real essence of structured data that provides utility to all i.e., profiles that include end-users, domain experts (including, real executives), systems analysts, systems integrators, programmers etc..

BTW – The U.S. Govt. is already making productive use of Linked (Open) Data based on the Entity Relationship Model. Naturally, that won't be changing anytime soon because such regression would be utterly illogical :-)

Excuse my typos, typed in haste en route to breakfast .

 **konklone** commented 15 days ago
Let's either merge or close this thing.

 **jcarbaugh** commented 15 days ago
@konklone +1

 **lmattels** commented 15 days ago
@konklone would it be democratic to close the most popular pull-request?

 **pmckinney** commented 15 days ago
@konklone +1 to merge or close. If people want to discuss the various issues raised here, there are more appropriate forums. In terms of whether to merge/close, I would opt to close, and maybe open a pull request to remove the "mosaic effect" link, to keep that list of links to non-contentious .gov links.

 **benbalter** commented 15 days ago
Wow. Awesome, awesome discussion that I can already tell is going to inform a lot of the project's decisions moving forward.

Sounds like there's a general consensus, as to @jqnatividad's original pull request, that linking out to additional linked data resources is a good arrow for agencies to have in their quiver.  to @konklone's suggestion to merge the pull request as is.

As for implementation / format, it sounds like there's a bunch of valid routes we can go. Would love for volunteers to shepherd breaking those ideas up into additional pull requests, so that we can create actionable, concrete steps that the community can recommend to agencies, even if it has to remain at a high level, or if we have to endorse multiple formats in parallel, until the broader linked data community settles on a standard.

Either way, this thread shows exactly why Project Open Data is so important... to hash out these types of opportunities, among those most invested with their resolution, before (hopefully) codifying as agency best practices. 

 **georgethomas** commented 15 days ago
@benbalter 

 **haleyvandyck** commented 15 days ago
Thank you to everyone on this thread for the very interesting conversation. @benbalter said it best-- this is exactly why we are excited about Project Open Data, and the opportunity this provides for continual iteration and improvement.

Looks like there is agreement on adding @jqnatividad suggested link.  Merging now.

Thank you all for contributing and we're looking forward to many more discussions and pull requests to come. 
- Haley, Senior Adviser at the White House

 **haleyvandyck** referenced this pull request from a commit 15 days ago
 **haleyvandyck** Merge pull request #21 from jqnatividad/LinkedDataCook 15 days ago 829a68e

 **haleyvandyck** merged commit 829a68e into project-open-data:master from jqnatividad:LinkedDataCook 15 days ago

 **haleyvandyck** closed the pull request 15 days ago



haleyvandyck commented

15 days ago



Thank you to everyone on this thread for the very interesting conversation. @benbalter said it best-- this is exactly why we are excited about Project Open Data, and the opportunity this provides for continual iteration and improvement.

Looks like there is agreement on adding @jqnatividad suggested link. Merging now.

Thank you all for contributing and we're looking forward to many more discussions and pull requests to come.

- Haley, Senior Adviser at the White House



haleyvandyck referenced this pull request from a commit

15 days ago

haleyvandyck Merge pull request #21 from jqnatividad/LinkedDataCook ...

029a68e

Merged

haleyvandyck merged commit 029a68e into project-open-data:master from jqnatividad:LinkedDataCook 15 days ago

case study #2

definition of an open license

[openhouseproject] New Memorandum almost defines open data, misses mark with open licenses



Inbox

x

-  **Josh Tauberer** My usual evaluation on principles... <http://razor.occams.info/blog/2013/05/09...> May 9 (6 days ago) 
-  **David Robinson** Does the federal government possess legal authority to apply a copyright lice... May 9 (6 days ago) 
-  **John Wonderlich** There is a funny tension between encouraging the use of "open licenses" (as t... May 9 (6 days ago) 
-  **Gunnar Hellekson** For the data that's in the public domain, my understanding is that the Federa... May 9 (6 days ago) 
-  **Gavin Baker** Federal government works are in the public domain (no copyright). If there's ... May 9 (6 days ago) 
-  **Eric Mill** Also, I know CC0 gets a lot of linkage, but something I discovered more recen... May 9 (6 days ago) 
-  **JAMES JACOBS** Thanks for this Josh. The law covering copyright of "government works" is pret... May 9 (6 days ago) 
-  **WashingtonWatch.com** I picked up on a theme that Josh alludes to, which is the de-emphasis of go... May 9 (6 days ago) 
-  **Josh Tauberer** A few more thoughts... That's exactly my fear, that in an attempt to appear t... May 9 (6 days ago) 
-  **Josh Tauberer** I'm usually one to say don't judge things on terms they weren't meant to be j... May 9 (6 days ago) 
-  **B.Klein** Been there, done that.....a program and infrastructure already exist for an ... May 10 (5 days ago) 
-  **Greg Elin** The below is my personal opinion... It would be great to have some get toget... May 10 (5 days ago) 
-  **Josh Tauberer** Except it modifies the definition.... May 10 (5 days ago) 
-  **Peggy Garvin** peggy@garvinconsulting.com via googlegroups.com to openhouseproje. May 10 (5 days ago) 

2:08 AM (21 hours ago) 



Gavin commented on how the phrasing "open licensing" in the Open Data Policy memorandum is "sub-optimal." Eric added: "Where CC0 is for **releasing** data into the public domain (implicitly indicating you could choose another license if you want), the Public Domain Mark indicates data is **already** in the public domain." And I see this as an edit suggested on Project Open Data.

My question: If a change to the policy is accepted on GitHub, will an amended OMB memorandum attachment be re-

ARE YOU COMING TO BED?

I CAN'T. THIS
IS IMPORTANT.

WHAT?

SOMEONE IS WRONG
ON THE INTERNET.



Joshua Tauberer's Blog

[Home](#) | [CV](#) | [Blog](#)

[« DC opens its “code”, embracing principles of open laws](#)

New Open Data Memorandum almost defines open data, misses mark with open licenses

TL;DR: The new E.O. and memorandum are good for transparency and lock in almost all of the generally accepted notions of open government data. But it misses the mark on the requirement of “open licenses.”

With an [executive order](#) and a new [Memorandum on Open Data Policy](#) today, the focus on entrepreneurship remained at the forefront of federal data policy. This focus began with last year’s Digital Government Strategy, and these days weather data and GPS signals are the examples of choice. That said, the policies set in the new memorandum are quite good for the classic use of this data (transparency, accountability, and civic education) even if “transparency” is only barely mentioned in passing.

Defining Open Data: How well does it do?

This new Open Data Memorandum presents the most detailed definition to date of “open data” by the federal government. It included many of the principles that our community has reached consensus on, but it gets one severely wrong.

As I wrote many years ago, the 2009 Open Government Directive

Author

For more about me, see [my home page](#).

Archives

- [» May 2013](#)
- [» April 2013](#)
- [» March 2013](#)
- [» January 2013](#)
- [» December 2012](#)
- [» November 2012](#)
- [» May 2012](#)
- [» April 2011](#)
- [» January 2011](#)
- [» December 2010](#)
- [» September 2010](#)
- [» July 2010](#)
- [» June 2010](#)
- [» February 2010](#)
- [» January 2010](#)
- [» December 2009](#)
- [» November 2009](#)
- [» September 2009](#)
- [» May 2009](#)
- [» April 2009](#)
- [» March 2009](#)

Conclusion

While the goals of the Memorandum in defining open data and using open licenses are laudable, the implementation does not meet the 8 Principles's requirements of open government data, at least under the usual understanding of "open license," and the use of the definition to promote open data is very limited.

PS. As Derek Willis [points out](#) over Twitter, the "mosaic effect" paragraphs in the memorandum are also somewhat concerning. The mosaic effect is hard to quantify and therefore difficult to limit, and this creates a big hole for keeping data government out of public reach.

UPDATE 5/10/2013 #1:

Rufus Pollock points out that the Open Data Commons Public Domain Dedication and License (PDDL) is similar to CCo and would also be appropriate. I agree.

Eric Mill notes that for data already in the public domain, the Creative Commons Public Domain Mark, which is basically an icon/badge, would be appropriate. Agencies should definitely mark public domain data as such.

UPDATE 5/10/2013 #2:

I added a few paragraphs to the section now called "*What's the definition used for?*".

This entry was posted on Thursday, May 9th, 2013 at 1:12 pm and is filed under [Civic Hacking](#). You can follow any responses to this entry through the [RSS 2.0 feed](#). You can [leave a response](#), or [trackback](#) from your own site.



This repository

Search or type a command



Explore Gist Blog Help



schacon



PUBLIC



project-open-data / project-open-data.github.io

Watch

Star

161

Fork

62



Code

Network

Pull Requests

22

Issues

38

Wiki

Graphs

Open

tauberer wants to merge 1 commit into project-open-data:master from tauberer:prose-patch-1

25 25 #64

Discussion

Commits 1

Files Changed 1



tauberer opened this pull request 15 days ago

Remove license examples that are not compliant with the memorandum's policy

No one is assigned

No milestone

The open data memorandum says that "[o]pen data are made available under an open license that places no restrictions on their use." Most open licenses place restrictions on use (such as attribution) and so would not be compliant with the memorandum's definition of open data. (See sections I, III-1-a, and III-1-c.)

This pull requests removes the examples that are not in line with the policy in the memorandum.

✓ Good to merge — The Travis CI build passed ([Details](#))

4 participants



tauberer added a commit

15 days ago



tauberer commented

15 days ago

See issues #5, #14.



tauberer commented

15 days ago



konklone commented 15 days ago

FWIW, I discuss this issue over at #5 and @philipashlock's PR #11. There are ways to address this, and I'd like to see something a touch stronger than just mentioning that government produced data is in the public domain.

As an example, the White House is currently using an explicitly bifurcated copyright notice [over at WhiteHouse.gov](#) for these situations. To the extent the licenses listed here need to reflect everything POD aims to cover, bifurcating the list into two lists seems like a fine approach,



dsmorgan77 commented 15 days ago

I am cool with treating the two issues separately, but this pull request does not do what you suggest, @konklone. I agree that the memorandum is inartfully silent on the issue of government-created data. But that doesn't invalidate the treatment of the issues for government-funded data.



tauberer commented 15 days ago

@dsmorgan77: There's no contradiction for the memorandum to apply to data acquired through a contract and for "no restrictions" to mean actually having no restrictions. That's perfectly plausible, and some would say good policy to boot.



dsmorgan77 commented 15 days ago

@tauberer sure, data acquired through a contract could be in the public domain. perfectly valid case, but not the standard the memorandum sets. and you're not addressing things like federally-funded research, where the administration has clearly set the bar at at least setting rights to require citation. so just deleting the licenses that are compatible with what the policy actually says isn't prudent or helpful to agencies that have to implement the policy.

i'm with @konklone. there's room for both treatments.



haleyvandyck commented 5 days ago

Thanks so much for the great discussion. We have our lawyers review all pull requests to the license file, and they are currently reviewing these edits. Thanks for your contribution!

- Haley, Senior Advisor at The White House



haleyvandyck commented

5 days ago (X)

Thanks so much for the great discussion. We have our lawyers review all pull requests to the license file, and they are currently reviewing these edits. Thanks for your contribution!

- Haley, Senior Advisor at The White House



CityCamp NC

@citycampnc



Follow

The White House gets it. Project Open Data.
"We can do more together than we can
alone." ow.ly/l6GGS #opendata #citycamp

[Reply](#) [Retweet](#) [Favorite](#) [More](#)



oisin @oisin

9 May

Dear Irish Gov, we, the citizens, think we deserve something like this
project-open-data.github.io

Followed by Chris Strom and 1 other

[Collapse](#) [Reply](#) [Retweet](#) [Favorite](#) [More](#)



Waldo Jaquith

@waldojaquith



Follow

So, U.S. CIO [@stevenvDC](#) just approved a
[@GitHub](#) pull request to retheme a federal
website. github.com/project-open-d...

[Reply](#) [Retweet](#) [Favorite](#) [More](#)



Open Government

@OpenGov



Follow

very cool: [@todd_park](#) and [@stevenvDC](#)
post free, open-source tools on GitHub to
accelerate new [#OpenData](#) policy: project-open-data.github.io

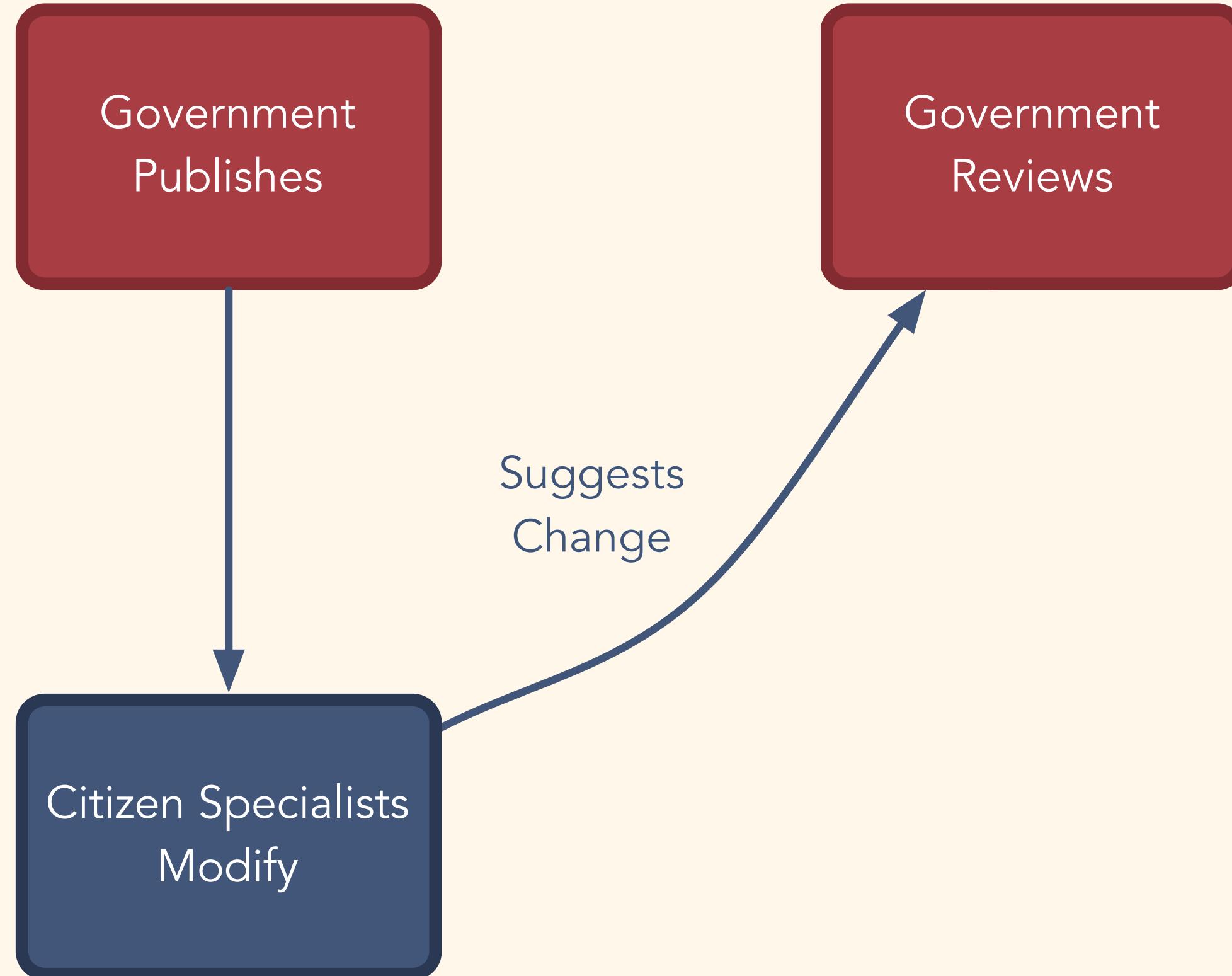
[Reply](#) [Retweet](#) [Favorite](#) [More](#)

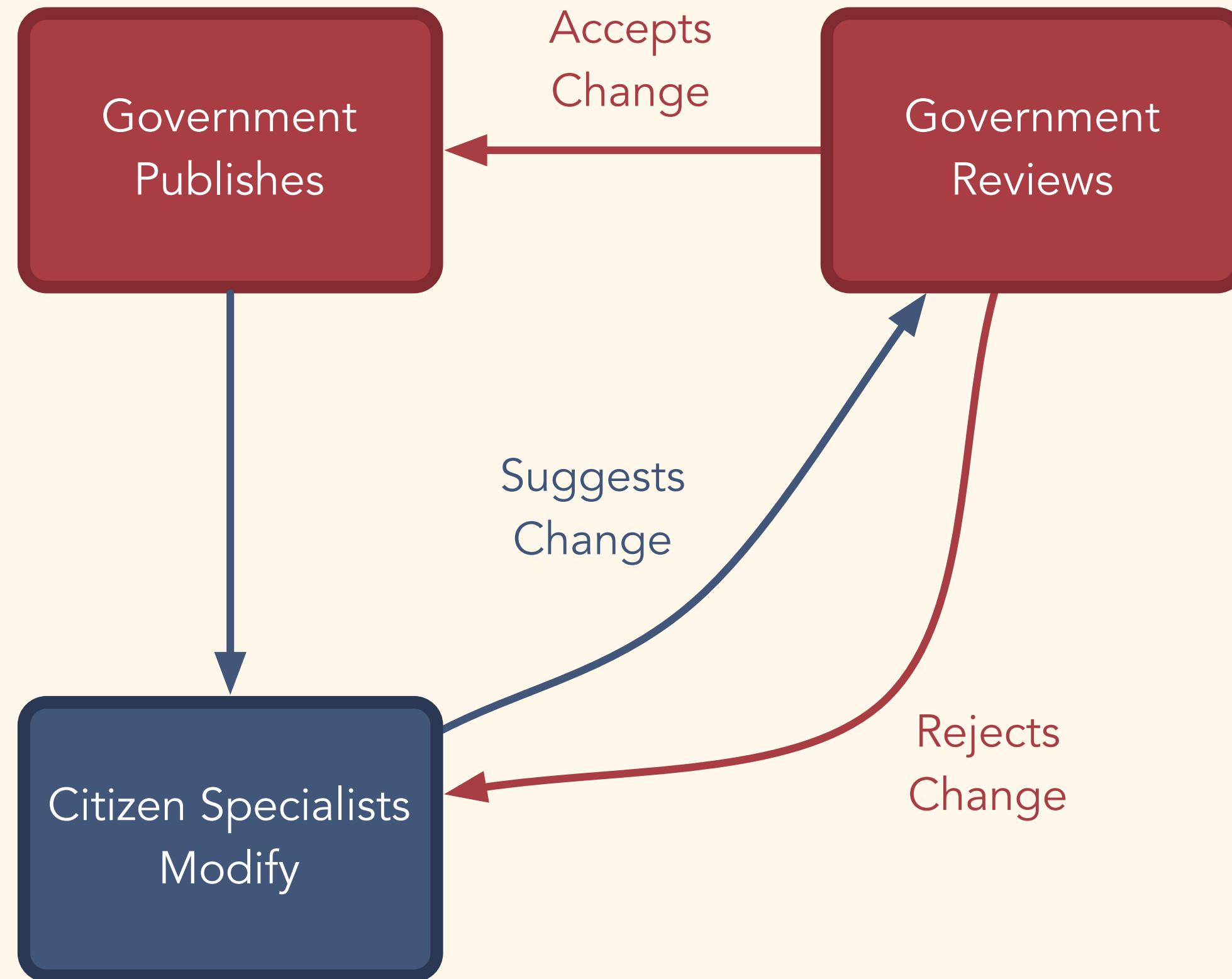
Government
Publishes

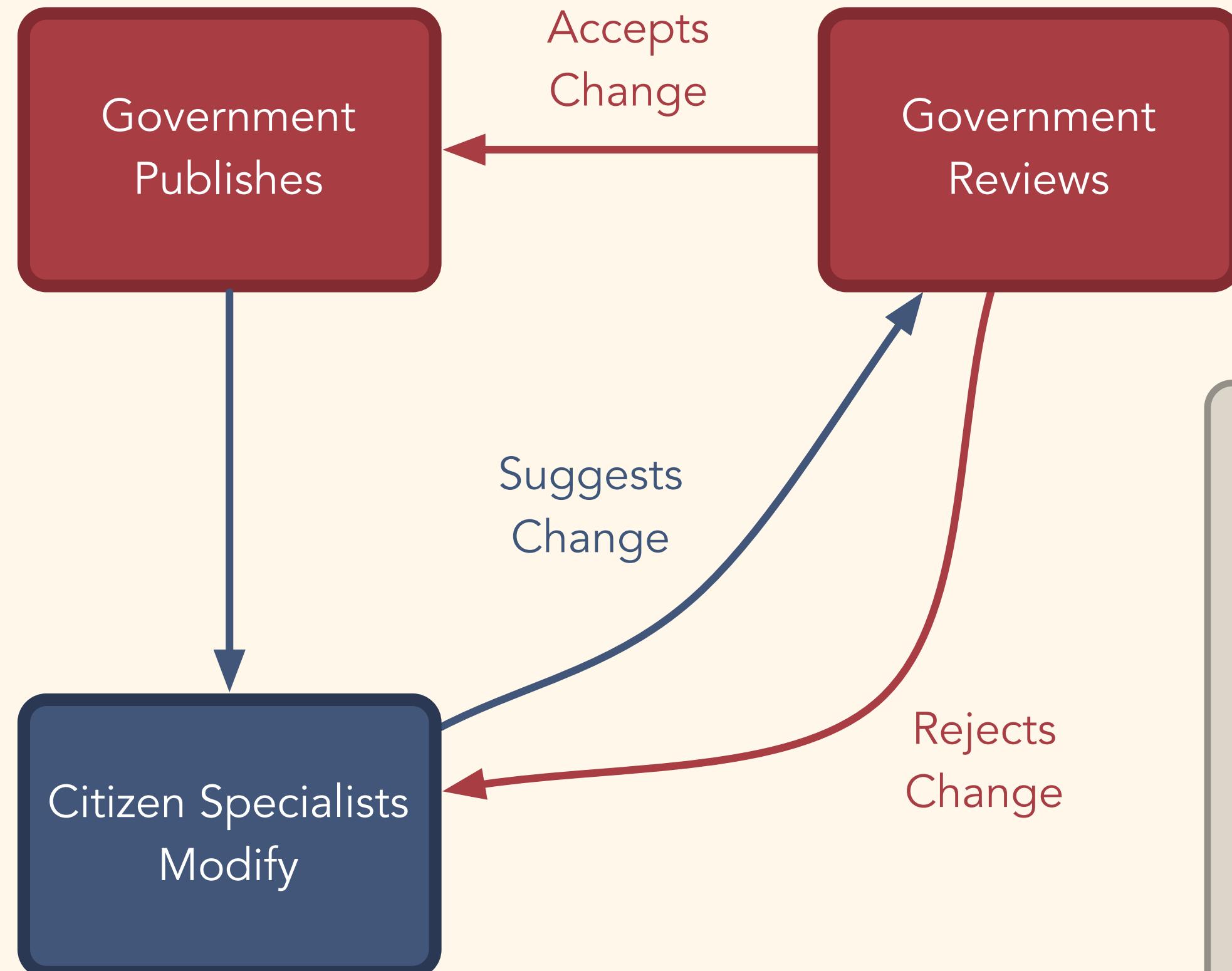
Government
Publishes



Citizen Specialists
Modify



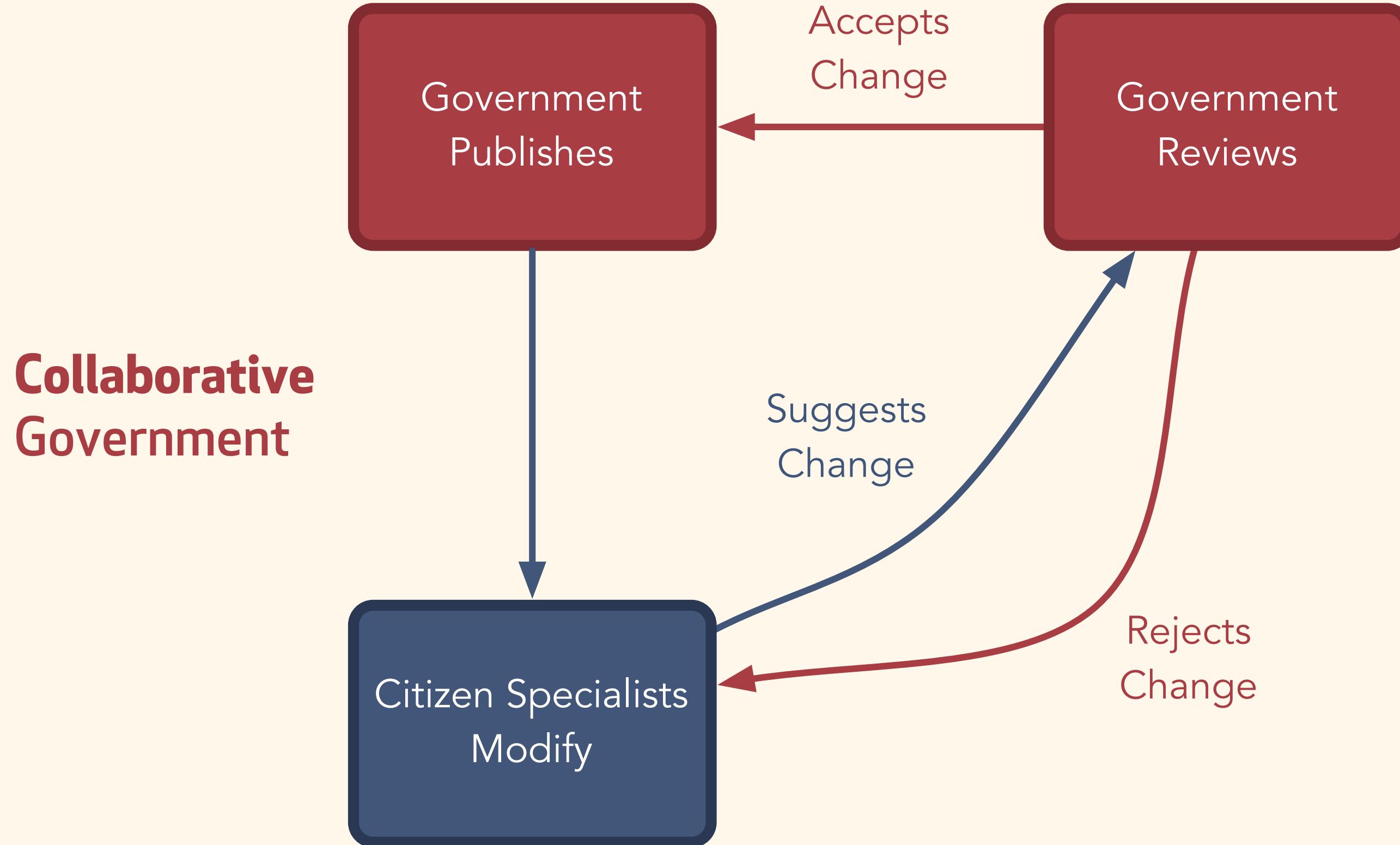




Data
Policy
Law
Budgets
City Planning

Government
Publishes

Open
Government



Collaborative Government

simple contribution model

clear path to implementation

"We see how politics, instead of being a dirty word, could be what it meant in the original Greek: the engagement of all citizens in the decisions that affect their lives."

Andrew Rasiej, PDF Opening Remarks, 2005

the future of your
civic contract

thank you

