Overview

Uniswap governance is in control of issuing licenses for production use of Uni V3.

Recently, a discussion around licensing Uni V3 for Arbitrum arose which gained a high level of community support. We immediately set to work on writing this proposal once the scope was described by the Uniswap team. The proposal is a suite of unit tests that locally forks hardhat, delegates enough votes to the Blockchain@UCLA for the test, and executes the following two actions:

1. Create the v3-core-license-grants

subdomain on uniswap.eth

, according to the Additional Use License specification as delineated in the UniswapBusiness Source License 1.1. * This is a one-time action: after the subdomain is created, further licenses may be issued by simply following the next step.

1. This is a one-time action: after the subdomain is created, further licenses may be issued by simply following the next step.

2. Set a text record defining the additional use license atv3-core-license-grants.uniswap.eth

In the process of writing this proposal, several questions about Uni V3 licensing in general arose in our minds, and this post seeks to elevate discussion around these points. These questions are not specific to Arbitrum licensing necessarily, so we decided to make a new topic for a more focused discussion.

Describing the permanence or impermanence of additional use grants

Regarding step 2. of the proposal, we did some light research (uhhh… Google-fu

) on what this means exactly. One thing of note is that with the way that ENS domains work, there can only be one record of a certain key

type at a subdomain at a given time. It's possible that subsequent licensing overwrites previous licensing, and it's possible that a subsequent update to the text records at this ENS domain might contradict previous records, eg attempt to revoke a previous additional use license.

One thing that isn't clear is if any Additional Use License is permanently forward-looking at the ENS domain

.

According to MariaDB, who defines the BSL used by Uniswap (which is referenced inV3 announcement post), licenses that are valid at one time are forward looking:

Q

: What if a licensor decides to change the use limitation in the future?

A

: A licensor may change the use limitation (i.e., by limiting or eliminating the Additional Use Grant) in the future, but users will always be able to use any previous version of the BSL software under its earlier limitations. In other words, the licensor's edits would be forward-looking only and would not apply retroactively.

According to that definition, any commercial use that abides by a previous licensing agreement, if it does so during the lifetime of that agreement, is valid forever, under that specific version of the license. However, if we are not mistaken, this applies specifically to the license defined here. Some clarity around this subject would be nice. We can see this going two ways:

1. Only licenses which are currently

live at the ENS address are considered valid. * This is possibly a compelling argument, because the Additional Use License says,

Additional Use Grant: Any uses listed and defined at

v3-core-license-grants.uniswap.eth

which makes no mention of permanence, impermanence, etc. Although, it could be construed that the past tense forms of "listed" and "defined" mean for any and all time, since "listed" and "defined" are actions that happened definitively in one moment and cannot be undone by the constraints of the linearity of time.

- In this case, it is important to define a process by which the Uniswap governance can append licenses without unintentionally revoking previous license grants, and it would be of interest to define the process by which intentionally revoking license grants would be valid, if that is possible and/or desired.
- This is possibly a compelling argument, because the Additional Use License says,

Additional Use Grant: Any uses listed and defined at

v3-core-license-grants.uniswap.eth

which makes no mention of permanence, impermanence, etc. Although, it could be construed that the past tense forms of “listed” and “defined” mean for any and all time, since “listed” and “defined” are actions that happened definitively in one moment and cannot be undone by the constraints of the linearity of time.

1. In this case, it is important to define a process by which the Uniswap governance can append licenses without unintentionally revoking previous license grants, and it would be of interest to define the process by which intentionally revoking license grants would be valid, if that is possible and/or desired.
2. Licenses granted as ENS text records even within a single block will always be valid, so long as they are issued during the validity of the BSL license at the proper time.
3. This is possibly a compelling argument, because ENS text records in even a single block live in the state of the blockchain indefinitely, and the additional use license doesn’t specify when

the listing is valid, so it might be fine to overwrite text records, since previous text records can be retrieved from archival nodes.

- In this case, it is important to create an interface which can query archival nodes for all text records ever set at the domain, so that UX around understanding licensing is as simple and transparent as possible.
- This is possibly a compelling argument, because ENS text records in even a single block live in the state of the blockchain indefinitely, and the additional use license doesn’t specify when

the listing is valid, so it might be fine to overwrite text records, since previous text records can be retrieved from archival nodes.

1. In this case, it is important to create an interface which can query archival nodes for all text records ever set at the domain, so that UX around understanding licensing is as simple and transparent as possible.

If we achieve clarity on this point, our recommendation would be to store the definition of the licensing agreement at the notice

key of the text record and to store licenses at the description

key.

Standardizing license grants

Another worthy mention: the exact wording of the license grant could be improved and possibly standardized into a template. If the community decides that it wants to create an onboarding process for deploying Uni V3 onto different platforms, then this would be desirable. As an example, xDAI has been requested, and we imagine that other scaling platforms will be requested. In contrast, if the community is of the mind that this is an activity which should be called sparingly, then it would make sense to specialize each particular license for each use case exclusively.

Safeguards against centralizing licensing

We identify a potential security risk regarding these additional use grants. It is possible that a malicious proposal passes which causes the subdomain v3-core-license-grants.uniswap.eth

to come under the control of a single entity. Depending on the permanence of additional use grants, this could be catastrophic in effect, or it can cause a years long stalemate until the license transitions to open source. As a safeguard, we imagine that placing the ENS subdomain in control of a hard-coded, specialized contract that is controlled by the governance, eg a contract with a 1 block delay before transfering the ENS domain, which would give the community some time to prevent such a malicious action from taking place, since a proposal could not atomically transfer it in that case, while maintaining Uniswap governance’s functional control of additional licensing. If an attacker were to attempt to steal the subdomain, then the delay between each proposal would give adequate time for the community to rally in response, because a follow up proposal would be subject to the same multi-day delays as the first one, despite only a 1 block delay being in the contract. This is because of each proposal having a span of days to vote and having the timelock’s execution delayed. Furthermore, Uniswap governance would still have a means to recover the address to any account, if desired, but it would take two proposals, which would be strong signaling of community support for that outcome in any case.

Blockchain@UCLA as a responsible steward of this proposal

In addition to these previous points, a veryvalid point was raised by the Uniswap team regarding Arbitrum licensing specifically:

If this action passed, any deploy that met the criteria could become the canonical V3 on Arbitrum. It might be desirable to coordinate this deployment process with the Uniswap Labs team, so that (for example) the factory addresses on all canonical deploys would continue to be identical, as well as the Arbitrum team, for any technical issues that may arise.

Hayden voiced that Uniswap would support the community in the event thatthis snapshot sentiment proposal passed, which it overwhelmingly did. Blockchain@UCLA has produced an EOA based proposal that would initiate this license issuance. Further, a contract based proposal was also written. So, we wrote them twice! Admittedly, the contract based proposal needs to have its unit tests updated to match the robustness of the EOA based proposal, and the EOA based proposal can do further checks to ensure that the proposal had its intended effect, ie query the v3-core-license-grants.uniswap.eth

ens address for its text record using the ENS javascript SDK with the forked hardhat provider instance. At this point, though, we feel we might be putting the cart before the horse, hence this discussion.

Blockchain@UCLA would love to be responsible stewards of this proposal, but we do not want to put either the Uniswap Labs team or the Offchain Labs team into a time crunch, and we do not want to initiate a community proposal that issues a grant before either team is ready to deploy, lest the Uni V3 deployment be maliciously front-run by a competitor. However, if we act soon, then we minimize the risk of this, because it's possible to pass this proposal before Offchain Labs opens Arbitrum to users, where the teams could collaborate meaningfully in the time reserved for developer onboarding, and where Offchain Labs could prevent other projects from deploying Uni V3 before Uniswap Labs team has time to coordinate the launch.

Final Summary / tl;dr

:

We bring up several points of discussion:

1. Describing the permanence or impermanence of additional use grants
2. Standardizing the LICENCE_GRANT.txt wording
3. Safeguards against adding a centralized controller to v3-core-license-grants.uniswap.eth

, which could potentially cut the Uniswap governance out of the ability to control licensing

1. Timing of deployment and whether Blockchain@UCLA is suitable as a responsible steward of this proposal