

Considering the Use of Walled Gardens for FLOSS Project Communication

Megan Squire^(✉)

Elon University, Elon, NC, USA
msquire@elon.edu

Abstract. At its core, free, libre, and open source software (FLOSS) is defined by its adherence to a set of licenses that give various freedoms to the users of the software, for example the ability to use the software, to read or modify its source code, and to distribute the software to others. In addition, many FLOSS projects and developers also champion other values related to “freedom” and “openness”, such as transparency, for example in communication and decision-making, or community-orientedness, for example in broadening access, collaboration, and participation. This paper explores how one increasingly common software development practice - communicating inside non-archived, third-party “walled gardens” - puts these FLOSS values into conflict. If communities choose to use non-archived walled gardens for communication, they may be prioritizing one type of openness (broad participation) over another (transparency). We use 18 FLOSS projects as a sample to describe how walled gardens are currently being used for intra-project communication, as well as to determine whether or not these projects provide archives of these communications. Findings will be useful to the FLOSS community as a whole as it seeks to understand the evolution and impact of its communication choices.

Keywords: Open source · Free software · Communication · Email · Mailing list · IRC · Stack overflow · Slack · Apache · Wordpress · Teams · Chat

1 Introduction

A common denominator between all free, libre, and open source software (FLOSS) projects is that they provide users with a software license that allows the user some level of freedom to read, modify, or distribute the software source code. Echoing these freedoms, FLOSS software is also produced in such a way as to foster openness and collaboration. For example, transparency in decision-making and welcoming participation are key values that are common to many FLOSS projects. These values have been called “open from day one” [1], or a “bazaar” style of organization [2], and have been attributed to the “success of open source” [3]. More recently the so-called “open source way” [4] is described as “a way of thinking about how people collaborate within a community to achieve common goals and interests” when applied to non-software contexts.

One software development practice that has traditionally been cited in the literature to preserve this openness is using publicly archived mailing lists for decision-making

and important project-related communication [5]. Mailing list archives preserve a **transparent** record of decision-making that can serve as an institutional memory and can help get new users up to speed quickly. Mailing lists also offer a **technological openness**, in other words a non-corporate-controlled, non-proprietary software system, ideally available under a FLOSS license. However, more recently, the FLOSS community has begun to ponder an additional perspective on openness: one that is defined by **inclusivity** and diversity of participation. [6–8] An industry publication recently bemoaned that older communication systems used in FLOSS (specifically IRC) are “complicated and unfriendly” and “the barrier to entry was a formidable challenge for the first time user” [9].

In this paper, we attempt to describe how one increasingly popular software development practice puts these openness values – openness through transparency and licensing, and openness through inclusivity – into conflict. Specifically, **communicating in “walled gardens”**, or non-open and corporate-controlled systems such as Slack or Stack Overflow, and **not keeping archives of this communication** puts the FLOSS goal of transparency into conflict with the goals of ease-of-use, inclusivity, and diversity of participation.

The remainder of this paper is organized as follows: first, we provide an overview of communication technologies used in FLOSS projects, and then we describe how a collection of 18 FLOSS projects currently relies on walled gardens for communication. For this, we use publicly available descriptions of existing FLOSS projects or repositories that are known to use walled gardens. By becoming aware of the size and scope of the practice of using walled gardens to communicate, the FLOSS community at large can choose how to react, including whether to embrace the practice, conduct additional research, take preventative measures, provide alternatives, or ignore the practice.

2 Communication Technology Used in FLOSS Projects

In keeping with the nature of FLOSS work as community-owned and community-driven, each individual software team makes the decisions about which communication technologies to use, and when to adopt or reject a new technology. Each team has its own requirements and makes its own determination of the positive and negative aspects of each communication choice. Here we describe two main types of technology, asynchronous and synchronous technologies, and how different FLOSS communities have used each one. For each category, we describe the alternatives in terms of the various “openness” values described previously: openness via transparency, openness via licensing and non-corporate control, and openness via inclusivity and ease-of-use.

2.1 Asynchronous Communication

Traditionally, many FLOSS communities have communicated using mailing lists. Some communities, such as the Apache project ecosystem, still require the use of mailing lists to conduct project business [10, 11]. There are several reasons for this preference. First, email is an **asynchronous** communication medium. Asynchronous communication

allows for messages that can be sent and read at different times. (Other examples of asynchronous communication include paper mail, email, bulletin board systems, and Web sites.) Asynchronous communication works especially well for FLOSS teams that may be geographically distributed, since messages can be sent and read at the convenience of both parties.

Another feature of email mailing lists that is helpful to FLOSS development is the ease of creating browsable, searchable mailing list **archives**. Feller and Fitzgerald write, “Archived discussions, which represent ‘self-documentation’ of projects, are critical in OSS development.”[5] Archives preserve a record of decisions and can help bring new contributors up to speed.

Finally, and significantly for many projects, generic email and mailing lists are **standards-based**, in that anyone can develop email software, and sending and receiving email requires no particular relationship or agreements with any single corporation. Email protocols and software are not owned or controlled by any one entity, corporate or otherwise. Generic email or mailing list systems can be contrasted with **proprietary**, but still asynchronous, systems such as the Google Groups web-based Usenet interface [12], or Stack Overflow, a web-based Question and Answer site increasingly used by many FLOSS projects to handle many kinds of technical support [13]. Colloquially, these closed, corporate-controlled systems are called **walled gardens**.

2.2 Synchronous Communication

Some FLOSS teams also elect to use **synchronous** communication technologies, such as chat or instant messaging, in which the users are communicating back and forth in real time. For example, FLOSS teams may conduct developer meetings using Internet Relay Chat (IRC) [14, 15]. Real-time chat systems, such as IRC (but also recently including new entrants into this space such as Rocketchat, Mattermost, Discord, or Slack), are also used to share ideas informally, to get immediate technical help, and to build camaraderie in the community [1]. Because of the ephemeral nature of chat, communities may not approach it with the same expectation of being a long-term archive as they would expect from an email mailing list. Still, some communities and IRC channels are archived, usually through the use of special archiving **bots**. One impressive example of chat archiving is the Ubuntu IRC log collection, which is available at <http://irclogs.ubuntu.com>. These archives cover discussions happening on nearly 300 different Ubuntu-related chat channels, starting in 2004.

As with email and asynchronous discussion systems, synchronous systems differ in whether they are a product of a single corporation, or whether they are a FLOSS-licensed or open protocol. For example, IRC is an application layer Internet protocol, and as such anyone can run a server or develop a client for it. In contrast, Slack (<https://slack.com>), is a synchronous chat system developed and operated by a corporation, and its rules about costs, archiving policies, data sharing, number of participants, and so on, are determined by the corporation alone. Slack has a single client, and a Terms of Service (ToS) that restrict its use. Slack is not FLOSS licensed. We therefore include corporate-controlled, non-FLOSS licensed synchronous messaging services such as Slack in our definition of **walled gardens**.

2.3 How FLOSS Values Conflict When Communicating in Walled Gardens

In 2015, FLOSS developer Drew Devault wrote a blog post entitled “Please don’t use Slack for FOSS projects” which argued that Slack is a walled garden, and any trend toward adopting it should be curtailed in favor of continuing with IRC which he says is “designed to be open”. [16] The comments section of this post illustrates the conflict between the value of open design on one hand, and the value of openness through ease-of-use and inclusivity on the other hand. In those 187 comments, the value of “openness” is invoked for both arguments. Similarly, the Wordpress project, in rationalizing their move to Slack for developer and user chat [17] gives six reasons for the move, and the first three of those have to do with the user interface: “Open for everyone, Friendly user interface, Easy asynchronous conversation”. With their invocation of “open for everyone” they are certainly referring to usability and not licensing, since Slack is not open source [18]. Interestingly, they also laud the ability of Slack to function in an “asynchronous” way, specifically contrasting it with IRC and Skype (which they call “real-time”). For this paper, we will continue to refer to Slack as a synchronous technology.

A related values conflict is whether FLOSS projects using walled gardens are being “open” (in the sense of transparent) if they do not provide archives of their communications. Should FLOSS projects need to provide archives of their communications, and do certain communication technologies make archiving easier or harder? In general, the asynchronous communication technologies like web pages and mailing lists are stored as files, and as such, will be easier to archive. FLOSS email mailing lists are usually archived both by the projects themselves, and archives for many projects are also available for search/browsing/downloading via third-party web sites such as MarkMail (<http://markmail.org>) and Gmane (<http://gmane.org>). Even though IRC is a synchronous communication medium, since it was invented in 1988, it has had many years to develop logging and archiving features, including a diverse set of archive bots. Text-based IRC logs are publicly available for many large projects including Ubuntu, OpenStack, Puppet, Perl6, many Apache Software Foundation projects, and so on. Projects using third-party synchronous walled gardens like Slack have the technical capability to produce text logs, but as we will discuss in the next section, do not typically do so.

In the next section we begin to describe the increasing use of walled gardens by 18 popular FLOSS projects, including whether or not the communications are archived, and what the community’s rationale is for using the walled garden.

3 Data on Walled Garden Usage in FLOSS Projects

The tables below show examples of FLOSS projects that have announced that they are using walled gardens as a primary communication channel. These tables focus on Slack as a walled garden since prior work already addressed the use of Stack Overflow for developer support [13], and because – as Sect. 4 will show – Stack Overflow’s “garden walls” are substantially lower and more porous than the walls surrounding Slack.

In the tables, URLs containing references to the evidence are provided in the Appendix as [A1], [A2], and so on. Table 1 contains information for a general collection

of FLOSS projects that rely on walled gardens for communication, and Table 2 contains information for only Apache Software Foundation (ASF) projects. We moved ASF projects into their own table so that they could be compared to each other, since they are all subject to the same rules about decision-making on mailing lists [10, 11].

Table 1. FLOSS projects using walled gardens for all or part of their communication

Community	Use of walled garden	Status of archives
Wordpress (all)	Moved from IRC to Slack. “Slack communication is used for contributing to the WordPress project, be it code, design, documentation, etc” [A1]	No consistent Slack archive. Occasional links to archives are posted (e.g. [A2]), but Slack login is required. The archives are not downloadable or searchable. IRC logs used to be available, but now only one channel is logged [A3]
Drupal (UX)	Uses Slack for “daily talk and weekly meetings” [A4]. Main site Drupal.org is still evaluating going to Slack in a two-year old thread still getting active comments [A5]	No Slack archive [A6]
Ghost	Users/devs “split between IRC and our forums” consolidated at Slack. Weekly meetings in Slack [A7]	Meeting summaries are available on [A8], but full logs require a Slack login
Socket.io	“Join our Slack server to discuss the project in realtime. Talk to the core devs and the Socket.IO community” [A9] [A10]	No Slack archive
Elementary OS	“we switched over to Slack from IRC/Google+ at ... in the early summer. It’s been a massive improvement.” [A11] No links to join Slack on public web site [A12]. Uses Stack Exchange for “common questions” [A13] [A14] [A15]	No Slack archive. No local Stack Exchange archive
MidoNet	“We recently saw some other communities moving [IRC] over to Slack, and decided to make the jump ourselves” [A16]	Uses Slackarchive.io for archives. [A17]
Reactiflux/ React.js	Moved from Slack to Discord after getting too big and Slack refused new invites. [A18] Still has Freenode IRC channel. Stack Overflow recommended for questions [A19]	No Discord archive. No local Stack Exchange archive
Bitcoin-core	Most discussion happens on IRC. Mentions Slack in passing [A20]	Uses Slackarchive.io for archives. [A21]

Table 2. Apache Software Foundation projects using walled gardens for all or part of their communication

Community	Use of walled garden	Status of archives
Apache Cordova	Users can “Join the discussion on Slack” [A22], which “is a replacement for IRC, but not a replacement for decisions and voting, that still needs to be on the list”[A23]	No Slack archive
Apache Groovy	“The Slack channel is not endorsed by the Apache Software Foundation, It’s run by Groovy enthusiasts in the community for casual conversations and Q&A. Official discussions must happen on the mailing lists only” [A24]	No Slack archive
Apache Hbase	Mailing lists still exist but “Our IRC channel seems to have been deprecated in favor of the above Slack channel” [A25] The Slack channel is only mentioned in Sect. 110.3 [A26] and 143.2 [A27] of the Reference Guide	No Slack archive
Apache Iota	“The user mailing lists ... is the place where users of Apache iota ask questions and seek for help or advice.... Furthermore, there is the [apache-iota] tag on Stack Overflow if you’d like to help iota users there.... You are very welcome to subscribe to all the mailing lists. In addition to the user list, there is also an iota Slack channel that you can join to talk to other users and contributors” [A28]	No Slack archive
Apache Kudu	Slack is where “developers and users hang out to answer questions and chat” [A29]	No Slack archive
Apache Mesos and Aurora	Developers and users hang out in ... Slack [A30] [A31] “Note that even if we move to Slack, we will make sure people can still connect using IRC clients and that the chat history is publicly available (per ASF guidelines)” [A32]	Mesos and Aurora both use Slackarchive.io for archives [A33]
Apache Spark	“For usage questions and help (e.g. how to use this Spark API), it is recommended you use the Stack Overflow tag apache-spark as it is an active forum for Spark users’ questions and answers” [A34]	No local Stack Overflow archive
Apache Spot	“Getting started” link on Apache Spot project page [A35] links to Github [A36] which states “If you find a bug, have question or something to discuss please contact us: –Create an Issue.... –Go to our Slack channel”	No Slack archive
Apache Thrift	Slack not officially mentioned on product pages, but team created and channel mentioned in one email thread [A37]	Uses Slackarchive.io for archives [A38]

The last column in each table shows whether the community is providing archives of the communication that happens in the walled garden. To determine whether archives were available, we performed the following procedure. First we searched for archives via the public web site for the project, and if those were not available, we searched for archives via Google, using the following queries:

- [community name/project name] slack
- [community name/project name] chat
- [community name/project name] archive
- [community name/project name] logs
- [community name/project name] slackarchive.io

With a few exceptions, most of the projects that did have an archive put the link to it in an obvious place, so the archives were easy to find.

These tables show that the majority of projects which *are* using walled gardens are *not* creating archives of these communications. In the next section we discuss some options for communities that *do* want to create archives.

4 Archiving Walled Gardens

If a community does decide to move to walled garden for communication, there are some strategies it can take to combat the potential for a corresponding loss of transparency. Creating archives of the communications – as would have been available with a mailing list or IRC channels – is one obvious and familiar solution. We will first discuss the options for creating archives of Slack, and then we will briefly address Stack Exchange/ Stack Overflow archiving.

4.1 Archiving Slack

There are a few different options for archiving Slack conversations, each of which have different positive and negative aspects. First, as we noted in Tables 1 and 2, there are third-party services, such as Slackarchive.io (<http://slackarchive.io>), which can create and host Slack archives. Slackarchive.io lists many open source projects on its “who is using” list, including Bitcoin-core, Midonet, Apache Mesos, and Apache Thrift. The archives are searchable and browsable by date, but the archives are not easily downloadable. There are no Terms of Service posted on the Slackarchive.io site, nor is there a robots.txt file. The archives themselves are displayed in a JavaScript-driven responsive web interface, making downloads inconvenient and non-trivial to automate.

Another option for creating archives for Slack is to connect it to IRC via the Slack bridge [19] or via a third party tool (e.g. Sameroom, available at <http://sameroom.io>), and once the chat is on IRC, the archives can be created there using an IRC archive bot. Depending on the client, IRC may or may not be able to understand advanced features of Slack, including direct messages, code formatting features, and document attachments. Users who choose to use IRC will not see these aspects of the Slack experience, nor will an IRC bot be able to archive them.

Third, community managers can take the approach of Wordpress and simply point people to the in-Slack archive, for example [A2]. The downsides of that approach are:

- Viewers of the archive must be signed in members of the channel.
- The archives are only browsable on a day-to-day basis (a “pick a date” widget is also available).

- By default, the archives are not searchable or downloadable by a non-administrative user.
- Some communities with a lot of messages in the archive have reported seeing errors reading, “Your team has more than 10,000 messages in its archive, so although there are older messages than are shown below, you can’t see them. Find out more about upgrading your team” [20].

4.2 Archiving Stack Exchange

The options for creating local archives of Stack Overflow and Stack Exchange sites are determined by a Creative Commons BY-SA 3.0 license [21] that allows reuse of Stack Exchange network data (for example questions, answers, and the like) as long as attribution rules are followed [22]. The site also periodically provides a CC-licensed Data Dump [23] with private identifying user data removed. Despite these generous terms, it does not appear that many FLOSS projects relying on Stack Overflow for developer or user support are creating their own archives of this data, nor are they providing context to Stack Overflow questions or answers from within their own ecosystems. Rather, the communities that are using Slack as a question-and-answer facility are simply pointing users to the relevant Stack Overflow tag or corresponding Stack Exchange subdomain.

5 Conclusion

This paper presents data on how 18 FLOSS projects (including 10 Apache Software Foundation (ASF) projects) use walled gardens to communicate with and between users and developers. We define walled gardens in terms of their ownership or control by a single corporation, as well as by their lack of FLOSS licensing to users. Examples of walled gardens include synchronous communication services like Slack, and asynchronous communication sites like Stack Overflow.

We posit that when walled gardens are chosen for communication, the community has decided to subjugate the FLOSS value of openness via transparent, non-corporate, FLOSS-licensed communication for a different - and equally compelling - definition of openness, namely an openness of easy participation and diverse contribution. One way that these competing values can both “win” is for the project to provide avenues for increased transparency after the walled garden is chosen, specifically by providing easy-to-find, publicly available, downloadable archives of the communication that happens inside the walled garden. This step would effectively open a “gate” into the walled garden and reassert the value of transparency once again.

Unfortunately, our data shows that only a handful of projects have made any attempt toward transparency by opening such a gate. This resistance to creating transparent archives of communication also persists in communities such as ASF that explicitly encourage archives and transparency in project communication.

By questioning the use of walled gardens for communication and evaluating their effects on multiple types of “openness”, we hope to begin a dialogue within the FLOSS community about how to preserve and extend its unique values.

Appendix

Below are the URLs referenced in Tables 1 and 2 as [A1], [A2], and so on.

- A1. <http://make.wordpress.com/chat>
- A2. <https://make.wordpress.org/polyglots/2015/04/16/chat-notes-https-wordpress-slack-com-archives-core/>
- A3. <https://irclogs.wordpress.org/>
- A4. <http://www.drupalux.org/tools-and-resources>
- A5. <https://www.drupal.org/node/2490332>
- A6. <https://www.drupal.org/node/2798167>
- A7. <https://blog.ghost.org/ghost-slack/>
- A8. <https://dev.ghost.org/public-dev-meeting-4th-october/>
- A9. <http://socket.io/slack/>
- A10. <http://rauchg.com/slackin/>
- A11. <https://news.ycombinator.com/item?id=8286291>
- A12. https://www.reddit.com/r/elementaryos/comments/25t7cg/where_to_talk_to_the_developers/
- A13. <https://elementary.io/support>
- A14. <https://elementary.io/get-involved>
- A15. <https://elementaryos.stackexchange.com/>
- A16. <https://blog.midonet.org/irc-chat-moved-slack/>
- A17. <http://midonet.slackarchive.io/>
- A18. <https://facebook.github.io/react/blog/2015/10/19/reactiflux-is-moving-to-discord.html>
- A19. <https://facebook.github.io/react/community/support.html>
- A20. <https://bitcoincore.org/en/contribute/>
- A21. <http://bitcoincore.slackarchive.io/>
- A22. <https://cordova.apache.org/contribute/>
- A23. <http://markmail.org/message/o6ltqszgeqykcu>
- A24. <http://groovy-lang.org/community.html>
- A25. <https://issues.apache.org/jira/browse/HBASE-16413>
- A26. <https://hbase.apache.org/book.html#getting.involved>
- A27. <https://hbase.apache.org/book.html#trouble.resources>
- A28. <https://iota.incubator.apache.org/contribute.html>
- A29. <https://kudu.apache.org/community.html>
- A30. <http://mesos.apache.org/community/>
- A31. <http://aurora.apache.org/community>
- A32. <https://lists.apache.org/thread.html/a1c53250a94d96e3f4038a76f93db01c3cc4d649df861f762373ac0f@%3Cdev.mesos.apache.org%3E>
- A33. <http://mesos.slackarchive.io/>
- A34. <http://spark.apache.org/community.html>
- A35. <http://spot.incubator.apache.org/>

- A36. <https://github.com/Open-Network-Insight/open-network-insight/tree/spot#contributing-to-apache-spot>
- A37. <https://www.mail-archive.com/dev@thrift.apache.org/msg32757.html>
- A38. <http://thrift.slackarchive.io/general/>

References

1. Fogel, K.: *Producing Open Source: How to Run a Successful Free Software Project*. O'Reilly and Associates, Sebastopol (2005). <http://producingoss.com>
2. Raymond, E.: *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Sebastopol (2001)
3. Weber, S.: *The Success of Open Source*. Harvard University Press, Cambridge (2004)
4. Red Hat Community Architecture Team: *The Open Source Way* (2009). <https://opensourceway.org/book>
5. Feller, J., Fitzgerald, B.: *Understanding Open Source Software Development*. O'Reilly & Associates, Sebastopol (2002)
6. Daniel, S., Agarwal, R., Stewart, K.J.: The effects of diversity in global, distributed collectives: a study of open source project success. *Inf. Syst. Res.* **24**(2), 312–333 (2013)
7. Kuechler, V., Gilbertson, C., Jensen, C.: Gender differences in early free and open source software joining process. In: Hammouda, I., Lundell, B., Mikkonen, T., Scacchi, W. (eds.) *OSS 2012*. IAICT, vol. 378, pp. 78–93. Springer, Heidelberg (2012). doi: [10.1007/978-3-642-33442-9_6](https://doi.org/10.1007/978-3-642-33442-9_6)
8. Vasilescu, B., Posnett, D., Ray, B., van den Brand, M.G., Serebrenik, A., Devanbu, P., Filkov, V.: Gender and tenure diversity in GitHub teams. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3789–3798 (2015)
9. Williams, O.: Slack is quietly, unintentionally killing IRC. *The Next Web* (2015). <http://thenextweb.com/insider/2015/03/24/slack-is-quietly-unintentionally-killing-irc/>
10. Apache Software Foundation (nd) Decision Making. <https://community.apache.org/committers/decisionMaking.html>
11. Apache Software Foundation (nd) Mailing Lists. <http://apache.org/foundation/maillinglists.html>
12. Braga, M.: Google, a Search Company, Has Made its Internet Archive Impossible to Search. *Motherboard* (2015). <http://motherboard.vice.com/read/google-a-search-company-has-made-its-internet-archive-impossible-to-search>. (Feb 13)
13. Squire, M.: Should we move to Stack Overflow? Measuring the utility of social media for developer support. In: *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015)*, vol. 2, pp. 219–228. IEEE Press (2015)
14. Shihab, E., Jiang, Z.M., Hassan, A.E.: On the use of Internet Relay Chat (IRC) meetings by developers of the GNOME GTK+ project. In: *Proceedings of the 2009 Mining Software Repositories*, pp 107–110 (2009)
15. Shihab, E., Jiang, Z.M., Hassan, A.E.: Studying the use of developer IRC meetings in open source projects. In: *Proceedings of Software Maintenance (ICSM 2009)*, pp 147–156 (2009)
16. Devault, D.: Please don't use Slack for FOSS projects (2015). <https://drewdevault.com/2015/11/01/Please-stop-using-slack.html>. Accessed 1 Nov 2015
17. Make Wordpress, Chat (nd) <https://make.wordpress.org/chat/>
18. Baker, J.: 4 open source alternatives to Slack. *Opensource.com* (2016). <https://opensource.com/alternatives/slack>

19. Slack Help Center (nd) Connect to Slack over IRC and XMPP. <https://get.slack.help/hc/en-us/articles/201727913-Connect-to-Slack-over-IRC-and-XMPP>
20. FreeCodeCamp: So Yeah We Tried Slack... and We Deeply Regretted It (2015). <https://medium.freecodecamp.com/so-yeah-we-tried-slack-and-we-deeply-regretted-it-391bcc714c81>, Accessed 21 June 2015
21. Creative Commons. Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) License. <https://creativecommons.org/licenses/by-sa/3.0/>
22. Stack Exchange Terms of Service. <http://stackexchange.com/legal/terms-of-service>
23. Stack Exchange, Inc.: Stack Exchange Data Dump (2016). <https://archive.org/details/stackexchange>. Accessed Dec 2016

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

