Project Name	...
Online team meeting	https://fau.zoom.us/j/63092613958
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2023ss01-apache-pulsar-ui
GitHub feature board	https://github.com/amosproj/amos2023ss01-apache-pulsar-ui
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/19
Team T-shirt (white)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/af483f4b-b0e1-4fd2-9447-08b0e173f5bc
Team T-shirt (black)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/f8fc2edd-668e-4223-815c-1014b1b5a8ed
Additional materials	...

Last Name	First Name	GitHub User Name	Email Address
Dreesens	Philipp	phildree	philipp.dreesens@gmail.com
Teschner	Niklas	nikkite99	niklas.teschner@web.de
Kafedzhieva	Emilia	emiliakaf	emilia.kafedzhieva@gmail.com
Schwarzmann	Sebastian	MPSebastian	sebi.schwarzmann@gmail.com
Haverkamp	Anna Lucia	ahaverkamp	anna.lucia.haverkamp@gmail.com
Tochman-Szewc	Julian Marcel	JulianTS	julian.tochmanszewc@gmail.com
Arnhold	Jonas	jnsrnld	jonasarnhold@web.de
Nasir	Shahraz	Shahraz98	nasirsharaz@gmail.com
He	Ziqi	iheziqi	heziqi4399@gmail.com

#	Meeting Day	Product Owner	Software Developer	Release Manager	Scrum Master	Comment
1	2023-04-19	Sebastian & Philipp	Everyone else	N/A	Emilia	
2	2023-04-26	Sebastian & Philipp	Everyone else	N/A	Emilia	
3	2023-05-03	Sebastian & Philipp	Everyone else	Niklas	Emilia	
4	2023-05-10	Sebastian & Philipp	Everyone else	Anna Lucia	Emilia	
5	2023-05-17	Sebastian & Philipp	Everyone else	Julian Marcel	Emilia	
6	2023-05-24	Sebastian & Philipp	Everyone else	Jonas	Emilia	
7	2023-05-31	Sebastian & Philipp	Everyone else	Ziqi	Emilia	Mid-term due
8	2023-06-07	Sebastian & Philipp	Everyone else	Shahraz	Emilia	
9	2023-06-14	Sebastian & Philipp	Everyone else	Niklas	Emilia	
10	2023-06-21	Sebastian & Philipp	Everyone else	Anna Lucia	Emilia	
11	2023-06-28	Sebastian & Philipp	Everyone else	Julian Marcel	Emilia	
12	2023-07-05	Sebastian & Philipp	Everyone else	Jonas	Emilia	
13	2023-07-12	Sebastian & Philipp	Everyone else	Shahraz	Emilia	
14	2023-07-19	Sebastian & Philipp	Everyone else	Ziqi	Emilia	Demo day!
15	2023-07-26	Sebastian & Philipp	Everyone else	N/A	Emilia	Retrospective

Goals	
	Creating a good product; Developing a useful open-source software;
	Create something cool and have fun
	Work as a team
	Provide a product that fulfills as many customer needs as possible
	Gain useful experience with the Scrum framework and Agile methodologies
Meeting norms	
	Answer yes or no question using Zooms built in feature „Reactions/Reaktionen“
	Following the scrum principle (review, retro, planning)
	Be on time or communicate if you are late
	Try to end the meetings on time
	Try not to interrupt people speaking (unless helpfull/necessary)
	Always be nice :)
Working norms	
	understandable code over wizardry => commenting
	Define and follow coding guidelines (including code- and commit-style, git-strategy and so on…) (enforce coding style using Git CI/CD?)
	Issues/Goal descriptions shall provide necessary information to fulfill the task in a valuable way
	Good documentations
	Code reviews
	Structured Commits, PRs, Issues ...
	Inform early on if some task cannot be done/fully completed within the defined Sprint
	Reviews are not personal (be objective)
Coordination norms	
	Be organized, ask help immediately if you are stuck
	Start planning as early as possible
	Continuously keep track of individual progress during the week
Communication norms	
	Everybody can bring on their point
	There are no stupid questions
	If there are team decision, everybody should at least communicate if they are fine with it
	Communicate early on if there is a problem, so that we have time to fix it together
	Be respectful to each other
Consideration norms	
	Everyone has a vote - majority wins? but before hear (+discuss?) arguments of either side
Cont. improvement norms	
	Focus on retrospective (constant feedback)
Rewards	
	Good grades as a team
Sanctions	
	Vote on sanction if necessary
	Communicate if there is a problem in e.g. the retro, so that the team can decide together what to do (and also that the person knows that there is a problem)

Ziqi He	
Julian Marcel Tochman-Szewc	
Anna Haverkamp	
Philipp Dreesens	
Emilia Kafedzhieva	
Sebastian Schwarzmann	
Jonas Arnhold	
Niklas Teschner	
Shahraz Nasir	

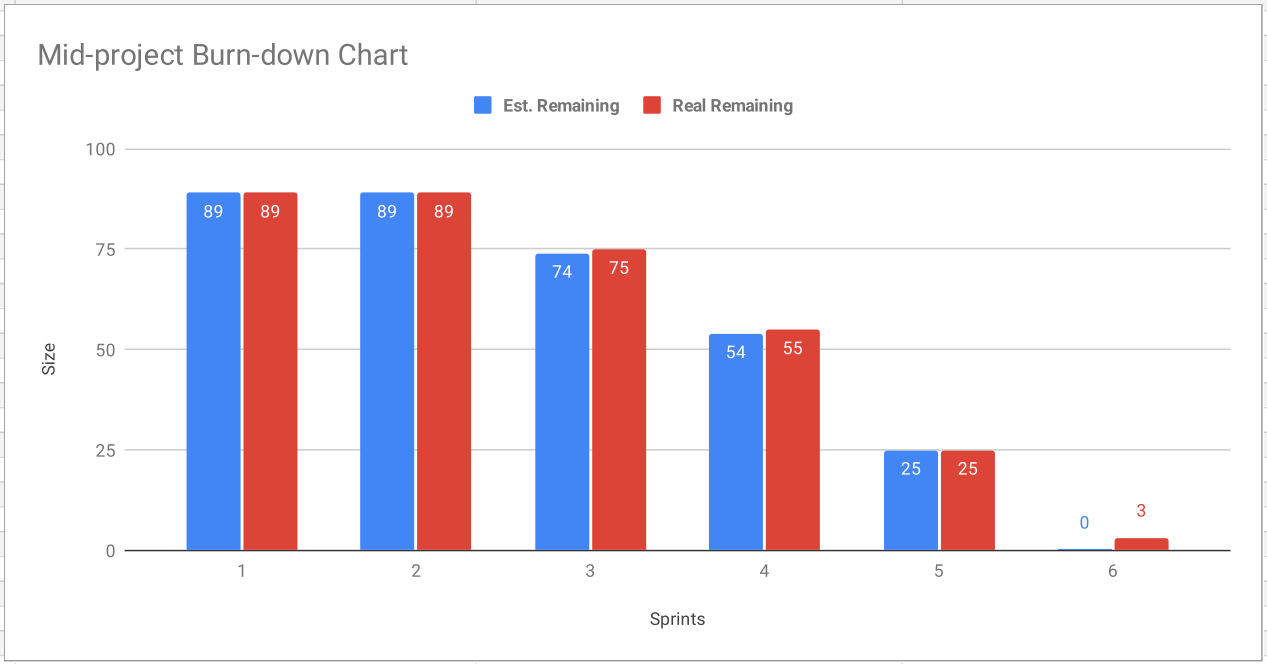
Product Vision	Project Mission
With increasing amounts of data shared across more and more complex and large IT-ecosystems, the need for a central coordination and communication platform like Apache Pulsar is increasingly important. Due to the large scale, such Apache Pulsar installations can become very complicated to overview and to maintain. Therefore, tools to handle Apache Pulsar installations like the Apache Pulsar Manager exist. Nonetheless, these tools do not provide sufficient support for the actual content exploration and manipulation, but instead mainly focus on the management of the topology. Our Apache Pulsar Web-UI should exactly fill this gap, by providing an intuitive way of navigating through an Apache Pulsar installation and exploring each topology level and its according (meta)data. The vision of our Web-UI is to support organizations that run Apache Pulsar, to increase transparency, maintainability and efficiency amongst their IT-infrastructure.	The mission of our project is well aligned with our product vision. Because the goal of our mission is to build a Web-UI that can easily be used by users that have some experience with managing and maintaining Apache Pulsar installations to understand and work on their infrastructure. We want to achieve this by structuring our UI according to the topology of Apache Pulsar, so that it can be navigated intuitively for targeted exploration of issues. With our Apache Pulsar UI issues can be located, and potential optimizations can easily be searched for, identified, and implemented.

Term	Definition
Stream	A continuous stream of messages, that can be analyzed, transformed and processed in real time
Topic	A logical entity or named channel in Pulsar through which messages are published and consumed. It represents a stream of data.
Partition	A division of a topic into multiple parallel streams of messages. Partitioning enables scalability and parallel processing across different consumers.
Producer	A client application that publishes messages to a topic in Pulsar.
Consumer	A client application that subscribes to a topic in Pulsar and consumes messages published to it.
Broker	A Pulsar component responsible for receiving and routing messages between producers and consumers. It handles the distribution and load balancing of messages across different topics and partitions.
Bookkeeper	A distributed storage system in Pulsar that provides fault tolerance and durability for storing message data.
Namespace	A namespace in Apache Pulsar is a logical grouping of topics. It acts as a container that isolates topics, allowing for better organization, access control, and resource allocation within a Pulsar cluster.
Tenant	A tenant in Pulsar represents a group or entity that has ownership over namespaces and topics. It provides multi-tenancy support, allowing different users or organizations to have their isolated environments within a Pulsar cluster.
Cluster	A Pulsar cluster consists of a group of interconnected brokers and bookkeepers that work together to provide the messaging and streaming capabilities. It forms the infrastructure for storing and processing messages.

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			89	89		
Sprints							
1	**Organizational and general research**			0	89	0	89
2	**Setup and Apache Pulsar research**			15	89	14	89
3	**Create basic Apache Pulsar Backend + Frontend**			20	74	20	75
4	**Improve UI and create Backend Interfaces**			29	54	30	55
5	**Integrate Tenants and Producers/Consumers**			25	25	22	25
6	**Integration of Backend and Frontend**			21	0	0	3
Features							
1	**Organizational and general research**						
		Get to know the team and talk about what we want to achieve					
			Schedule a meeting with the industry partner and the whole team	-		-	
			Get to know Apache Pulsar	-		-	
			Design team t-shirt	-		-	
			Design a team logo	-		-	
2	**Setup and Apache Pulsar research**						
		Kickoff project by setting up the boilerplate code and researching about apache pulsar					
			Plan and create the Software Architecture PDF	3		3	
			Project Setup with React/Angular JS	2		3	
			Project Setup with Spring	2		2	
			Get to know Apache Pulsar	3		2	
			Research APIs to interact with Apache Pulsar (type script)	3		3	
			Research and setup CI/CD Pipeline	2		1	
3	**Create basic Apache Pulsar Backend + Frontend**						
		Start experimenting with an apache pulsar instance and create basic web UI. Gather information/inspiration from other (software) products in this field					
			Create a new topic in Apache Pulsar	3		3	
			Send message in given topic	3		3	
			Retrieve raw message data from Apache Pulsar	3		3	
			Display all existing topics from Pulsar	2		2	
			Local Apache Pulsar Environment	3		3	
			Create basic topic/message frontend	3		3	
			Apache Pulsar Manager (APM) research	3		3	
4	**Improve UI and create Backend Interfaces**						
		Improve UI to become more user friendly and provide backend endpoints that can be used by the UI to retrieve the information that has to be shown					
			Create UI Design	2		2	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Create Landingpage of Apache-Pulsar-UI	2		3	
			Create base UI component	5		3	
			Create Cluster View	3		5	
			Create Namespace View	5		5	
			Message Informaion Interface	3		3	
			Namespace Information Interface	3		3	
			Cluster Information Interface	3		3	
			Topic Information Interface	3		3	
5	**Integrate Tenants and Producers/Consumers**						
		Extend UI according to our newly obtained knowledge about apache pulsar and clean up the backend code					
			Create "Build Process Video	-		-	
			Create Consumer Information Popup	3		2	
			Create Producer Information Popup	3		2	
			Create Message View	5		5	
			Create Topic View	3		3	
			Create Tenant View	2		2	
			Add Tenants to Navbar	1		1	
			Tenant Information Interface	3		2	
			Refactoring Backend	5		5	
6	**Integration of Backend and Frontend**						
		Make UI use the information provided by the backend in order to show real data to the user					
			Create build/deploy Documentation	-		-	
			Remove Welcome Page	1			
			Producer Infromation Interface	2			
			Consumer Information Interface	2			
			Make Frontend use Backend API	8			
			Import Dummy Data into our Apache Pulsar Environment	5			
			Check Functionality for Mid-Project Release	3			

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining



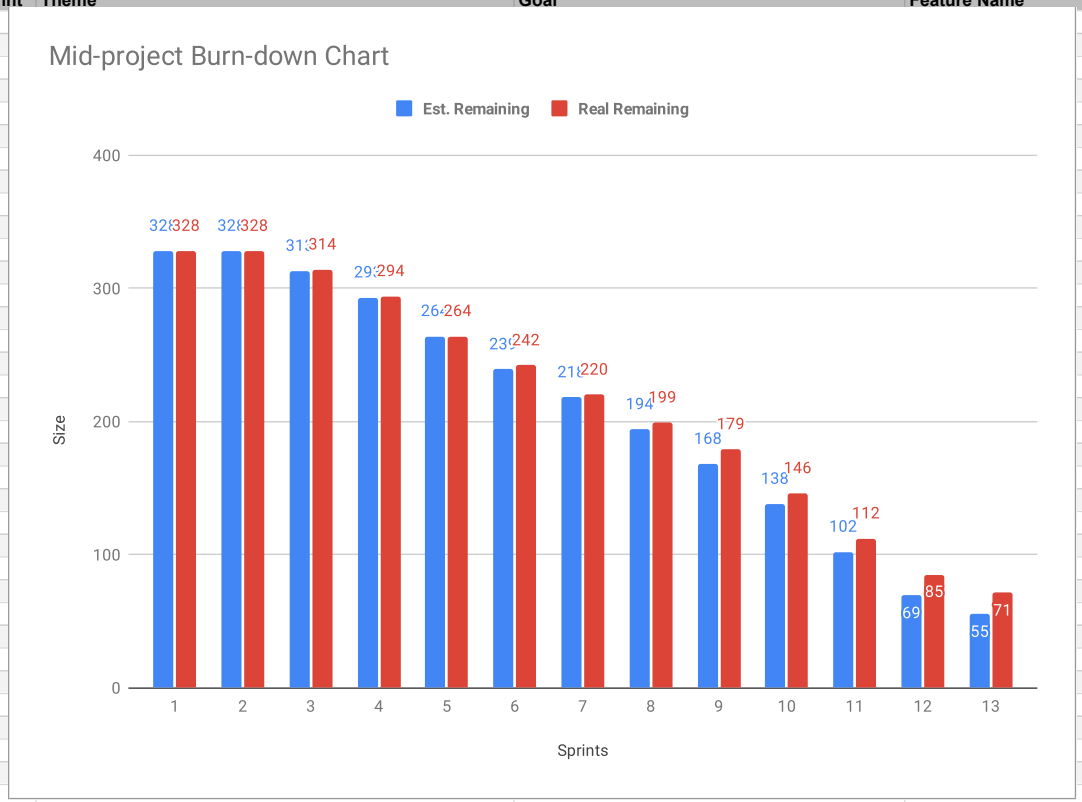
Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			328	328		
Sprints							
1	**Organizational and general research**	Get to know the team and talk about what we want to achieve		0	328	0	328
2	**Setup and Apache Pulsar research**	Kickoff project by setting up the boilerplate code and researching about apache pulsar		15	328	14	328
3	**Create basic Apache Pulsar Backend + Frontend**	Start experimenting with an apache pulsar instance and create basic web UI. Gather information/inspiration from other (software) products in this field		20	313	20	314
4	**Improve UI and create Backend Interfaces**	Improve UI to become more user friendly and provide backend endpoints that can be used by the UI to retrieve the information that has to be shown		29	293	30	294
5	**Integrate Tenants and Producers/Consumers**	Extend UI according to our newly obtained knowledge about apache pulsar and clean up the backend code		25	264	22	264
6	**Integration of Backend and Frontend**	Make UI use the information provided by the backend in order to show real data to the user		21	239	22	242
7	**Integrate Consumer/Producer/Subscription**	Extend UI at topic level to include consumers, subscriptions and producers		24	218	21	220
8	**Refactoring for better integration and Bugfixing**	Restructure endpoints in order to enable better usage from the frontend in the future and adapt frontend to new endpoints & minor bugfixes in frontend		26	194	20	199
9	**Provide more Info from Backend and Display more Info**	Continue impoving data/information that is retrieved in the backend (from Apache Pulsar) and finalize the integration of the new endpoints in the frontend (+ new/improved parts of UI)		30	168	33	179
10	**Test Scalablilty and Improve UI**	Use the AWS credentials from the customers to host a VM, where we can create more than thousand topics and test the scalability of our product		36	138	34	146
11	**Scalability Adjustments and Finalizing UI**	Improve the UI partially by cleaning some things up (and improving the design) and also by implementing solutions we found for problems identified during the scalability testing		33	102	27	112
12	**Minor refinements of UI, Backend and Setup**	Simplify deployment of the project (by creating a single docker-compose file and an CORS interface) and implement minor improvements towards usability and functionality of the frontend (better info text, cache flushing, deduplicated messages)		14	69	14	85
13	**Refactoring, Updating Documentatoins and Demo Day**	During this sprint we mainly focus on refactoring, updating the documentations and creating a nice demo day story for an awesome demo day		55	55	29	71
Features							
1	**Organizational and general research**						
		Get to know the team and talk about what we want to achieve					
			Schedule a meeting with the industry partner and the whole team	-		-	
			Get to know Apache Pulsar	-		-	
			Design team t-shirt	-		-	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Design a team logo	-		-	
2	**Setup and Apache Pulsar research**						
		Kickoff project by setting up the boilerplate code and researching about apache pulsar					
			Plan and create the Software Architecture PDF	3		3	
			Project Setup with React/Angular JS	2		3	
			Project Setup with Spring	2		2	
			Get to know Apache Pulsar	3		2	
			Research APIs to interact with Apache Pulsar (type script)	3		3	
			Research and setup CI/CD Pipeline	2		1	
3	**Create basic Apache Pulsar Backend + Frontend**						
		Start experimenting with an apache pulsar instance and create basic web UI. Gather information/inspiration from other (software) products in this field					
			Create a new topic in Apache Pulsar	3		3	
			Send message in given topic	3		3	
			Retrieve raw message data from Apache Pulsar	3		3	
			Display all existing topics from Pulsar	2		2	
			Local Apache Pulsar Environment	3		3	
			Create basic topic/message frontend	3		3	
			Apache Pulsar Manager (APM) research	3		3	
4	**Improve UI and create Backend Interfaces**						
		Improve UI to become more user friendly and provide backend endpoints that can be used by the UI to retrieve the information that has to be shown					
			Create UI Design	2		2	
			Create Landingpage of Apache-Pulsar-UI	2		3	
			Create base UI component	5		3	
			Create Cluster View	3		5	
			Create Namespace View	5		5	
			Message Informaion Interface	3		3	
			Namespace Information Interface	3		3	
			Cluster Information Interface	3		3	
			Topic Information Interface	3		3	
5	**Integrate Tenants and Producers/Consumers**						
		Extend UI according to our newly obtained knowledge about apache pulsar and clean up the backend code					
			Create "Build Process Video	-		-	
			Create Consumer Information Popup	3		2	
			Create Producer Information Popup	3		2	
			Create Message View	5		5	
			Create Topic View	3		3	
			Create Tenant View	2		2	
			Add Tenants to Navbar	1		1	
			Tenant Information Interface	3		2	
			Refactoring Backend	5		5	
6	**Integration of Backend and Frontend**						

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Make UI use the information provided by the backend in order to show real data to the user					
			Create build/deploy Documentation	-		-	
			Remove Welcome Page	1		1	
			Producer Infromation Interface	2		2	
			Consumer Information Interface	2		3	
			Make Frontend use Backend API	8		8	
			Import Dummy Data into our Apache Pulsar Environment	5		5	
			Check Functionality for Mid-Project Release	3		3	
7	**Integrate Consumer/Producer/Subscription**						
		Extend UI at topic level to include consumers, subscriptions and producers					
			Update Views UI to show General/Detail Information	8		5	
			Refactoring Frontend	5		5	
			Remove Welcome Page	1		1	
			Update Producer Information Endpoint	2		2	
			Update Subscription Endpoint	2		2	
			Update Endpoints to provide Gernal and Detailed Information	5		5	
			Cleanup Branch Management	1		1	
8	**Refactoring for better integration and Bugfixing**						
		Restructure endpoints in order to enable better usage from the frontend in the future and adapt frontend to new endpoints & minor bugfixes in frontend					
			Restructuring Backend Endpoints	8		8	
			Error Handling & Propagation	5		5	
			Adapt frontend to use new backend Endpoints	5		-	
			BUGFIX: filter selection in conjuction with drill downs	3		2	
			BUGFIX: Navbar does not cleanup filter	2		2	
			Create Consumer Information Endpoint	3		3	
9	**Provide more Info from Backend and Display more Info**						
		Continue impoving data/information that is retrieved in the backend (from Apache Pulsar) and finalize the integration of the new endpoints in the frontend (+ new/improved parts of UI)					
			Implement Message Endpoint	3		3	
			Add Schema to Topic Details Endpoint	2		2	
			Update Producer / Consumer Information Popup	3		3	
			Create Subscription Information Popup	3		3	
			Adapt frontend to use new backend Endpoints	5		13	
			Add Producer/Subscriptions to Topic View Filtering	8		8	
			Update Topic-View (Consumers -> Subscriptions)	1		1	
			Detect/Display Filter Collisions	5		-	
10	**Test Scalablilty and Improve UI**						
		Use the AWS credentials from the customers to host a VM, where we can create more than thousand topics and test the scalability of our product					
			Improve UI Scaling	5		5	
			Make use of Screen Real Estate	3		3	
			Setup AWS VM to host a persistent Apache Pulsar Instace	8		8	
			Use VM to test the Scalability of our Project	3		2	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Adapt Python Scripts from Customer to create Apache Pulsar Testing Topology	5		5	
			Remove global Search	1		1	
			Show specific aggregations in each overview	5		5	
			Add Schema to Topic Details in the UI	1		2	
			Research if Message Endpoint is able to provide Messages for {Producer, Subscri	5		3	
11	**Scalability Adjustments and Finalizing UI**						
		Improve the UI partially by cleaning some things up (and improving the design) and also by implementing solutions we found for problems identified during the scalability testing					
			Extend message endpoint with parameters for producer and subscription	3		3	
			Adjust Filtering UI	5		2	
			Improve Element Card regarding redundant Information and Styling	3		3	
			Show some more Information in the Views Titles	3		3	
			Implement Topic Drilldown to Messages	5		5	
			Reorder Popup content and show messages	3		3	
			Links in elements should be useable for navigation	3		3	
			Improve Performance of Preview Elements	8		5	
12	**Minor refinements of UI, Backend and Setup**						
		Simplify deployment of the project (by creating a single docker-compose file and an CORS interface) and implement minor improvements towards usability and functionality of the frontend (better info text, cache flushing, deduplicated messages)					
			Adjust docker-compose file	3		3	
			Set minimum value of shown messages to 1	1		1	
			Remove duplicated Messages in the Backend	2		2	
			Adjust CORS-Policy in the Backend	3		3	
			Enable Flush Functionality in the Frontend for the user	2		2	
			Create info text	3		3	
			Create Demo Day Video	-		-	
			Create Demo Day Slide	-		-	
13	**Refactoring, Updating Documentatoins and Demo Day**						
		During this sprint we mainly focus on refactoring, updating the documentations and creating a nice demo day story for an awesome demo day					
			Refactoring Frontend: Add missing JSDoc components	3		3	
			Prepare a conclusive demo day story	5		8	
			Refactoring Backend: Seperating Producer and Subscription Service	5		5	
			Refactoring Backend: Remove Consumer Service	3		3	
			Refactoring Backend: Class Building Process	2		2	
			Refactoring Frontend: Introduce Topology Enum	3		3	
			Fix Docker Compose Setup on AWS	2		2	
			Research Fix for Issues with the Mocking Framework	3		3	
			Finalize user/design/build documentation	-		-	
			cleanup and finish final project release plan	-		-	
			tag code with final-project-release	-		-	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining



Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Feature is done when the acceptance criteria are fullfiled	Release is done when all test pass	Project is done when code builds and runs on all target plattforms
	Feature is done when there are test where it is necessary and they pass	Release is done when there are no more necessary fixes	Project is done when the documentation provides good guidances to user on how to use the tool
	Feature is done when the linter passes	Release is done when all team members are happy the current state	Project is done when the tool does what it is supposed to do and the major requirements are fullfiled (customer is sastisfied)
	Feature is done when it is merged onto our main branch (after the code has been reviewed by at least one other person)		
	Feature is done when the code has been documentend or commented properly		

Type	Link / reference

#	Context	Name	Version	License	Comment
	Backend dependencies				
1	Java Spring framework	Spring Boot	2.7.11	Apache License Version 2.0	
2	Java testing framework	JUnit	5.8.2	Common Public License	
3	Java library	lombok	1.18.26	MIT License	
4	Pulsar Java library	Pulsar Client	2.11.1	Apache License Version 2.0	
5	Integration Testing	Testcontainers	1.18.0	MIT License	
6	Documentation	Springdoc-Openapi	1.7.0	Apache License Version 2.0	
7	Caching	Caffeine	2.9.3	Apache License Version 2.0	
	Frontend dependencies				
8	User Interface	React	18.2.0	MIT License	
9	React UI tools	Material UI	5.12.2	MIT License	
10	React state management	Redux Toolkit	1.9.5	MIT License	
11	Data fetching	Axios	1.4.0	MIT License	
12	Code formatter	Prettier	2.8.8	MIT License	
13	Linter	ESLint	8.39.0	MIT License	
14	Testing Framework	Jest	27.4.3	MIT License	
15	Masonry Component	React Plock	3.0.2	MIT License	
16	CSS Framework	Tailwind CSS	3.3.2	MIT License	
17	React routing	React Router	6.12.1	MIT License	
18	CSS Pre-processor	sass	1.62.1	MIT License	
19	SASS/SCSS Loader	sass-loader	13.2.2	MIT License	

Last Name	First Name	Value					
				3.00	OK		
Teschner	**Niklas**	3					
Haverkamp	**Anna Lucia**	3		0	No size		
Tochman-Szewc	**Julian Marcel**			1	Trivial size		
Arnhold	**Jonas**	3		2	Small size		
Nasir	**Shahraz**			3	Medium size		
He	**Ziqi**			5	Large size		
				8	Very large size		
				13	Too large (size)		