# Introduction

The topic of this thesis is the application of scientific approaches to lower the initial hurdle for Small and medium-sized Enterprises (SMEs) to move from their existing non-web desktop legacy systems to modern web-based systems. The introduction motivates the need for a dedicated approach to address the intial unwillingness and resistance of these companies to commence a Web Migration. High effort and risk due to the lack of a dedicated approach supporting the intial phase of a Web Migration are identified as main challenges. This thesis contributes to the solution of these problems by providing a set of methods and technical infrastructure to support initiation of Web Migration.

0 Ref

l[ink](https://www.tu-chemnitz.de/)

asdfasdf

## Situation

### A h3 heading

#### A h4 heading

Web Applications are widely used and familiar for users, as they are offering advantages over traditional desktop applications such as instant deployment, a common standards-based target development platform and a high level of interactivity. While there are different definitions for the term *Web Application* (Kappel, Pröll, and Reich 2006; Kienle and Distante 2014), in this thesis we follow the definition of TODO:CITE.

TODO:vs Web System and Web Site (Kienle and Distante 2014)

Definition according to (Kappel, Pröll, and Reich 2006)

A Web Application is a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser.

Definition: Web Application

A Web Application is a software system which has a Client-Se rver architecture, uses HTTP(S) as main communication protocol and that uses Web Browsers for rendering HTML on the client side.

One of the main characteristics of Web Engineering - the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web Applications (Deshpande et al. 2002) - is the continous evolution of Web technologies (Gitzel, Korthaus, and Schader 2007). Modern Web Applications are the result of this evolution (Kienle and Distante 2014) which started with the proposal of Hypertext as Information Management system by Tim Berners Lee in March 1989 (Berners-Lee 1990). Quickly evolving from the original textual content-focused system, the Web started to be perceived as “a universal, standards-based integration platform” (Knorr 2003). Technologies for dynamic behavior on the client and server side led to the introduction of the term *Web Application*, sometimes also called Rich Internet Application (RIA), indicating that their user experience became similar to the one of desktop applications (Kienle and Distante 2014). Increased usage of dynamic content allowed users to use the Web as mass collaboration system in the era refered to as Web 2.0 in the mid 2000s. TODO:CITE

Driven by a buisness perspective with focus on envisioned cost savings and increased flexibility (Kienle and Distante 2014), composing applications from *Web Services*, i.e. re-usable pieces of functionality invokeable via HTTP, became a focus in both research and industry. The architectural paradigm of Service-oriented Architecture (SOA) TODO:CITE is closely related to this development and an ecosystem of web protocols such as Web Service Description Language (WSDL) for describing the service interface TODO:CITE, SOAP as messaging protocol TODO:CITE and BPEL for orchestrating web services TODO:CITE was created.

The success of the Web Service paradigm was continued in the advance of *Cloud Computing*, allowing web applications to be built based on scaleable sets of rapidly provisioned, shared resources. TODO:CITE Kienle 585 (Cloud computing is not exclusively about web applications, but is has influenced them a lot) While the related paradigms of Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) mainly influenced the architecture of web applications, the Software as a Service (SaaS) paradigm - the most largest segment of the public cloud market with increasing tendency according to (Statista 2018a) also impacted end users’ perception and expectations of software (Politis 2017) allowing ubiquitous access without installation.

In a similar way, the high popularity and use of mobily devices in private TODO:CITE and work contexts (Bitkom 2013) changed end users’ expectations with regards to high levels of interactivity and social interactions (Bitkom 2013). This influenced web applications in two ways: the need for open application programming interfaces (APIs) for integration with third party systems and along with it the creation of corresponding protocols like OAuth and the requirement of flexible presentation layers capable of handling the vast amount of different screen sizes, resolutions and touch-based interactions giving rise to responsive design frameworks like Bootstrap TODO:CITE.

(Evolution Chain (Kienle and Distante 2014) all steps have been relevant targets for migrations due to advantages increased complexity, made it more difficult for SMEs)

The above evolution of web applications has brought various advantages over traditional desktop applications, such as: - Instant Deployment - Standardized target development platform - Low access requirements - High level of Interactivity and Social Interactions

The web’s underlying client-server architecture allows for instant deployment by enabling to perform updates in one place (Gitzel, Korthaus, and Schader 2007). This is either directly on the server, if the locus of code execution is on the server side, or indirectly on the server, if the locus of code execution is on the client side but the code is distributed via the server. The ability to instantly deploy new versions is of great advantage for companies, since it allows for shorter development cycles and thus faster reactions on changing or new requirements and reduced time-to-market (Khadka et al. 2014).

In contrast to the wide range of proprietary technologies and frameworks for traditional desktop applications, web applications enable companies to focus on one single target development platform based on open standards. This significantly reduces the development workload that is required to provide a software for many different platforms on many different devices, since only one version of the software based on one set of technologies is needed. Additional effort and cost reductions are achieved by only requiring one development infrastructure and set of tools, as opposed to the platform-wise toolchains (e.g. compilers, IDEs) of desktop applications. Open web standards like HTTP, HTML, CSS and JavaScript form the core of web applications. They are standardized, published and continously updated by international standardization bodies like IEEE and W3C and can be publicly accessed and used without restrictions. The risk of technology deprecation is significantly lower than for proprietary desktop application technologies TODO:CITE.  
Development staff needs only be trained on one technological base and can be hired from a potentially larger group of developers with experience in web technologies.

From the end users’ perspective, web applications impose low requirements for accessing software compared to desktop applications. According to (Politis 2017), 86% of end users find SaaS applications help them succeed more than desktop alternatives. They can be used from any system with a web browser. No installation is required, reducing time and complexity for users. The browser has become a standard interface (Aversano et al. 2001) / standard client (Gitzel, Korthaus, and Schader 2007) for many users to access software. Web browsers are available for almost any device and platform used by end users. The resulting high *application portability* (Gitzel, Korthaus, and Schader 2007) fosters a ubiquity of software that allows for new work patterns like home office (cf. SaaS-Powered Workplace, 38% of companies almost entirely running on SaaS as of 2017 according to (Politis 2017), (Bitkom 2013). The lower complexity and effort gives software companies access to address potentially larger users bases.

In particular, social interactions in potentially large user bases have become widespread in web applications for end users, allowing formation of instant online communities (Bressler and Grantham 2000) which that are in turn advantageous for companies through higher levels of user engagement. Also in working contexts, social information sharing is advantageous (cf. Shareconomy (Bitkom 2013)).

On the technological level, this is enabled by two main factors. The use of URLs as uniform handles for addressing web resources and application states enables quick sharing of arbitrary web application contents and has seen wide integration into modern platforms (cf. Sharing functionalities of mobile operating systems) and applications (cf. link-based sharing in most SaaS applications such as Dropbox, Office365 or GitHub). The availibility of APIs for third party developers and corresponding web protocols like OAuth allow better integration with other web applications.

Due to these advantages, web applications are becoming dominant over desktop applications, with new applications being built as web applications (Politis 2017) and existing applications being replaced by web applications (Pettey and Meulen 2012). From a business perspective, the aforementioned cloud computing paradigm of SaaS has seen the most interest (Statista 2018a). While SaaS comprises more aspects than software architecture and is thus no a synonym for web applications, the shift from traditional on-premise software to SaaS-based cloud software is closely related to web applications since the required interfaces and communication are mostly implemented using standard web protocols and mos SaaS solutions are accessed via HTTP and use browsers as clients. Thus, the following SaaS-related studies are also indicative of tendencies of usage of traditional on-premise desktop applications versus web applications.

The total size of the public cloud SaaS market was already 91.75 billion USD worldwide in 2016 (Statista 2018b), and is growing (Statista 2018b; Politis 2017; Chan 2018) The cloud shift rate of the SaaS market in 2016 to 2020 is growing three times faster than PaaS and two times faster than IaaS. (Politis 2017) The number of SaaS applications that organizations use is rising, many companies will be running purely on SaaS soon. (Politis 2017) Software vendors react on these developments by no longer building traditional on-premise software. What is more, 86% of end users say SaaS applications help them succeed more than desktop alternatives. (Politis 2017) Existing solutions are becoming not only extended by but increasingly replaced with SaaS solutions (Pettey and Meulen 2012).

Shifting perspective away from the above business perspective towards evidence from end users, studies show that acceptance of web applications is high even in traditional sectors like banking. For instance, online banking is well-established for internet users in Germany (Pols et al. 2016).

In spite of the widespread use and familiarity of web applications for users and their advantages over traditional desktop applications as described above, there are still many non-web legacy systems existing and modernization remains an important topic in industry (Batlajery et al. 2014). TODO:CITES FROM BELOW

Legacy systems and legacy modernization have been an interest for research for a long time. Several definitions exist in literature. An overview can be found in (Wagner 2014). In this thesis, we follow

(Maybe only characteristics instead of one definition)

Definition Legacy Information System (Bisbal et al. 1999)

Any information system that significantly resists modification and evolution.

Legacy systems are characterised as follows:

* they are business critical, i.e. they represent great value for the company due to the huge amount of knowledge about business processes, rules etc. (Aversano et al. 2001; Sneed, Wolf, and Heilmann 2010; Wagner 2014)
* they resist modification and evaluation (Bisbal et al. 1999)
* they are poorly documented or have a complete lack of documentation (Sneed, Wolf, and Heilmann 2010; Warren 2012; Batlajery et al. 2014)
* there is a lack of experienced manpower in the company which developed the system (Batlajery et al. 2014) due to changes or retirement of the original staff
* they are based on obsolete technologies that are incompatible with current/future technological environments and potentially discontinued (Pérez-Castillo et al. 2013; Batlajery et al. 2014; Heil and Gaedke 2016)
* Too rigid to comply with new business requirements (Batlajery et al. 2014)

Surveys among practitioners from software industry confirm these characteristics (Batlajery et al. 2014).

* Legacy System Definition
  + Reasons for their existence (Khadka et al. 2014) (business-critical, perceived as reliable/stable/proven – never touch a running system, optimized performance)
* TODO: find statistics (<https://docs.google.com/document/d/1iZpumRpQrOuquwLusJk8VzgOS_T1p9oOn1v5nYKWrCo/edit>)
* Gartner: legacy modernization at position 5 of top ten buisness priorities, 2013

<https://www.gartner.com/newsroom/id/2304615>, Previous years: <http://www.gartner.com/imagesrv/cio/pdf/cio_agenda_insights2013.pdf>

* NASCIO. State CIO Top Ten Policy and Technology Priorities for 2017, position 5: <https://www.nascio.org/Portals/0/Publications/Documents/2017/NASCIO-TopTen-2017.pdf>
* Forrester: Application Modernization: Procrastinate At Your Peril! 2011

Mayority of industry does not use academic resources for modernization (Batlajery et al. 2014)

* Technical Debt topic attracts attention in academia and industry
  + Technical debt definition
  + CAST Report on Application Software Health 2011
  + Estimating the Principal of an Application's Technical Debt, IEEE Software.
  + A systematic mapping study on technical debt and its management, Journal of Systems and Software
  + How do software development teams manage technical debt? – An empirical study, Journal of Systems and Software

##Problem statement

* Making a transition from non-web legacy systems to web is desirable for companies (“widely recognised as a must for keeping competitive in the dynamic business world” (Aversano et al. 2001))
* BUT: Commencing web migration is hard
  + Effort
    - High effort (TODO: Distinguish Big Bang/Incremental (Distante et al. 2006)reports 12+PM for one project, (Bernhart et al. 2012)18+ PM for incremental approach, (Aversano et al. 2001) 8PM, (Maenhaut et al. 2016) reports about 14 PM for two projects), not only including direct development, but also training for users etc.
    - Hard to estimate, cf. Corrective Maintenance Estimation (Singh 2015) and difficult to predict the ROI (Khadka et al. 2014)
  + Risk
    - Uncertainty of success/risk of failure due to feasibility threats (staff not trained in migration, not trained in web development, LS typically complex)
    - Uncertainty of desirability (acceptance by customers, usability of web-based solution)
* (Existing migration approaches assume decision to migrate is taken already)
* Existing migration approaches do not support initial phase
  + Cf. phase support from (Heil and Gaedke 2017)

##Question

* How to support companies (SMEs?) to commence web migration

##Solution

* Define web migration process addressing initial fears/resistance
* Risk minimization through migration pilots (prototypes)
* Demonstrate desirability and feasibility of web version of legacy system
* Identify and maintain existing valuable knowledge through concept assignment platform (annotation platform?)
* Apply rapid prototyping to web migration
* Control impact of web migration on customers through measuring visible changes

##Contributions

* Decision Support System for SMEs
* Concept Assignment Process integrated into development and Platform
* Rapid Migration Prototyping
* Similarity Metrics for non-web and web UIs

Aversano, Lerina, Gerardo Canfora, Aniello Cimitile, and Andrea De Lucia. 2001. “Migrating legacy systems to the Web: an experience report.” In *Proceedings Fifth European Conference on Software Maintenance and Reengineering*, 148–57. IEEE Comput. Soc. <https://doi.org/10.1109/.2001.914979>.

Batlajery, B V, R Khadka, a M Saeidi, S Jansen, and J Hage. 2014. “Industrial Perception of Legacy Software System and their Modernization.” September. <https://servicifi.files.wordpress.com/2014/02/finaltechrep3.pdf>.

Berners-Lee, Tim. 1990. “Information Management: A Proposal.” <http://info.cern.ch/Proposal.html>.

Bernhart, Mario, Andreas Mauczka, Michael Fiedler, Stefan Strobl, and Thomas Grechenig. 2012. “Incremental reengineering and migration of a 40 year old airport operations system.” In *2012 28th Ieee International Conference on Software Maintenance (Icsm)*, 503–10. IEEE. <https://doi.org/10.1109/ICSM.2012.6405313>.

Bisbal, Jesús, Deirdre Lawless, Bing Wu, and Jane Grimson. 1999. “Legacy information systems: issues and directions.” *IEEE Software* 16: 103–11. <https://doi.org/10.1109/52.795108>.

Bitkom. 2013. “Arbeit 3.0: Arbeiten in der digitalen Welt.” Berlin: BITKOM. [http://www.bitkom.org/de/publikationen/38338{\\_}77013.aspx](http://www.bitkom.org/de/publikationen/38338%7b\_%7d77013.aspx).

Bressler, Stacey E., and Charles Grantham. 2000. *Communities of Commerce: Building Internet Business Communities to Accelerate Growth,Minimize Risk,and Increase Customer Loyalty*.

Chan, Viva. 2018. “60 SAAS Statistics That Will Change the Way You Think.”

Deshpande, Yogesh, San Murugesan, Athula Ginige, Steve Hansen, Daniel Schwabe, Martin Gaedke, and Bebo White. 2002. “Web Engineering.” *Journal of Web Engineering* 1 (1). Rinton Press: 4—–17.

Distante, Damiano, Gerardo Canfora, Scott Tilley, and Shihong Huang. 2006. “Redesigning legacy applications for the web with UWAT+: A Case Study.” In *Proceeding of the 28th International Conference on Software Engineering - Icse ’06*, 482–91. New York, New York, USA: ACM Press. <https://doi.org/10.1145/1134285.1134353>.

Gitzel, Ralf, Axel Korthaus, and Martin Schader. 2007. “Using established Web Engineering knowledge in model-driven approaches.” *Science of Computer Programming* 66 (2): 105–24. <https://doi.org/10.1016/j.scico.2006.09.001>.

Heil, Sebastian, and Martin Gaedke. 2016. “AWSM - Agile Web Migration for SMEs.” In *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*, 189–94. Enase. Rome, Italy: SCITEPRESS - Science; and Technology Publications. <https://doi.org/10.5220/0005869301890194>.

———. 2017. “Web Migration - A Survey Considering the SME Perspective.” In *Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering*, 255–62. SCITEPRESS - Science; Technology Publications. <https://doi.org/10.5220/0006353502550262>.

Kappel, Gerti, Birgitt Pröll, and Siegfried Reich. 2006. *Web Engineering – The Discipline of Systematic Development of Web Applications*. Wiley.

Khadka, Ravi, Belfrit V. Batlajery, Amir M. Saeidi, Slinger Jansen, and Jurriaan Hage. 2014. “How do professionals perceive legacy systems and software modernization?” In *Proceedings of the 36th International Conference on Software Engineering - Icse 2014*, 36–47. New York, New York, USA: ACM Press. <https://doi.org/10.1145/2568225.2568318>.

Kienle, Holger M., and Damiano Distante. 2014. “Evolving Software Systems.” In *Evolving Software Systems*, edited by Tom Mens, Alexander Serebrenik, and Anthony Cleve, 1st ed., 201–28. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-45398-4>.

Knorr, Eric. 2003. “2004: The Year of Web Services.” <https://www.cio.com/article/2439869/web-services/2004--the-year-of-web-services.html>.

Maenhaut, Pieter-Jan, Hendrik Moens, Veerle Ongenae, and Filip De Turck. 2016. “Migrating legacy software to the cloud: approach and verification by means of two medical software use cases.” *Software: Practice and Experience* 46 (1): 31–54. <https://doi.org/10.1002/spe.2320>.

Pettey, Christy, and Rob van der Meulen. 2012. “Gartner Survey Shows 71 Percent of Respondents Using SaaS for Less Than Three Years.” Gartner. <http://www.gartner.com/newsroom/id/2253215>.

Pérez-Castillo, Ricardo, Ignacio García-Rodríguez de Guzmán, Ismael Caballero, and Mario Piattini. 2013. “Software modernization by recovering Web services from legacy databases.” *Journal of Software: Evolution and Process* 25 (5): 507–33. <https://doi.org/10.1002/smr.1554>.

Politis, David. 2017. “2017 State of the SaaS-Powered Workplace.” BetterCloud. <https://www.bettercloud.com/monitor/wp-content/uploads/sites/3/2017/05/2017stateofthesaaspoweredworkplace-report-1.pdf>.

Pols, Axel, Katja Hampe, Franz Grimm, and Christopher Meinecke. 2016. “Digital Banking.” Berlin: Bitkom Research. <https://www.bitkom.org/NP-Marktdaten/Konsum-und-Nutzungsverhalten/E-M-Commerce/Studienbericht-Digital-Banking-2014-bis-2016-161104.pdf>.

Singh, Satwinder. 2015. “Estimating Effort for Corrective Software Maintenance Using Neuralnetworks and Regression Technique.” *International Journal of Computer Science and Communication Engineering* 4 (1).

Sneed, Harry M., Ellen Wolf, and Heidi Heilmann. 2010. “Einleitung zum Thema Softwaremigration.” In *Software-Migration in Der Praxis: Übertragung Alter Softwaresysteme in Eine Moderne Umgebung*, 1st ed., 1–23. dpunkt Verlag.

Statista. 2018a. “Cloud Software as a Service - Statistics & Facts | Statista.” Statista. <https://www.statista.com/topics/3071/cloud-software-as-a-service-saas/>.

———. 2018b. “Total size of the public cloud Software as a Service (SaaS) market from 2008 to 2020 (in billion U.S. dollars).” <https://www.statista.com/statistics/510333/worldwide-public-cloud-software-as-a-service/>.

Wagner, Christian. 2014. “Fundamentals.” In *Model-Driven Software Migration: A Methodology*, 9–66. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-05270-6>.

Warren, Ian. 2012. *The renaissance of legacy systems: method support for software-system evolution*. Springer Science & Business Media. <https://doi.org/10.1007/978-1-4471-0817-7>.