IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 6, 2017, accepted November 6, 2017, date of publication November 17, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2775042

An Introduction to Dew Computing: Definition, Concept and Implications

PARTHA PRATIM RAY
Department of Computer Applications, Sikkim University, Gangtok 737102, India
ppray@cus.ac.in

ABSTRACT Since the end of the 1990s, the world has witnessed a tremendous growth in the area of information and communication technology (ICT), starting with grid computing, cloud computing (CC), and fog computing to recently introduced edge computing. Although, these technologies are still in very good shape, they do heavily rely on connectivity, i.e., Internet. To address this challenge, this paper proposes a novel dew-cloud architecture that brings the power of CC together with the dew computing (DC). Originally, the dew-cloud architecture is an extension of the existing client-server architecture, where two servers are placed at both ends of the communication link. With the help of a dew server, a user has more control and flexibility to access his/her personal data in the absence of an Internet connection. Primarily, the data are stored at the dew server as a local copy upon which instantiation of the Internet is synchronized with the master copy at the cloud side. Users can browse, read, write, or append data on the local dew site, which is a local Web form of an actual website. With the incorporation of the dew domain naming system and dew domain name redirection, mapping between different local dew sites has become possible. Novel services, such as infrastructure-as-a-dew, software-as-a-dew service, and software-as-a-dew product, are, hereby, introduced along with the DC. This paper presents the following as key contributions: 1) a precise and concrete definition of DC; 2) detailed and comprehensive discussions of its concept and working principle; 3) application potentials; and 4) technical challenges. The motto of this paper is to conceptualize the fact of empowerment of the ICT-user base with almost an Internet-free surfing experience in coming days.

INDEX TERMS Dew-cloud architecture, dew server, dew site, dew database, DDNS, DDNR, dew service.

I. INTRODUCTION

Several computing paradigms have remarkably changed the way of ''web surfing'' experience in past two decades. Grid, cloud, fog, and edge computing represent the key pillars of this evolution. With incorporation of smart phones and high-speed communications, inter-network access has been reached a new level of sophistication. Multiple cloud domains and fog services are currently engaged into providing of required set of data or information to its users or customers. Favorable aggregations of service oriented aspects are presently acting as the basis of such interventions. Users are indebted towards data storage, analysis, visualization, computation, and persuasion as per the prescribed notions of the cloud, fog or edge venders. Subsequently, network users are getting heavily dependent on the availability of internet connectivity to persuade for the ''opted'' jobs. Thus, resulting into the origination of a stringent environment of despot cyber-statesmanship around the virtual world. A revision in existing computing models is henceforth expected to cope-up with provisional current-time issues, as identified.

This paper presents an introductory view of a novel computing paradigm Dew Computing (i.e. DC). DC is envisaged to rely upon ''micro-services'' approach in highly heterogeneous, vertical, and distributed manner. It opens up a new window of centralized-virtualization-free computational horizon, whereby providing a scope for scattering of multi-typical data into the low-end devices. Thus, it creates the novel opportunity toward data accessibility without continuous internet availability. As a result, DC is indebted towards covering up all the existing network technologies, wide range of key characteristics (e.g. cooperative application, independent, mobile data aggregation etc.) and hybrid network behavior. The significant advantages of DC are apprehended as follows: self-healing attribute, autonomic self-augmentation, self-adaptive, transparent, user-programmability, super user experience, and extreme scalability. Despite of having highly heterogeneous

device pool, DC is capable of performing complex tasks in its own vicinity. To serve such functionality, DC needs an advanced modular architecture that shall conform all related aspects in DC-ecosystem.

The contributions of this paper are as follow:

- To propose a novel definition of Dew Computing; for better understandability;
- To propose a novel Dew-Cloud architecture;
- To elaborate the theories, key concepts and working behind DC;
- To present potential applications and issues to be resolved.

This paper is organized as follows. Section II elaborates key concepts behind grid, cloud, fog, and edge computing paradigms. Section III compares several existing definitions of DC and propose a novel one. Section IV presents theories and in-depth concepts of DC. Section V discusses on application potentials with respect to a novel 4-tier architecture. Section VI discussed on possible research challenges and prospects. Section VII concludes the paper.

II. BACKGROUND OF GRID, CLOUD, FOG AND EDGE COMPUTING

Before going into the major topic of discussion, let's get started with the grid, cloud, fog, and edge computing, as follows.

A. GRID COMPUTING

Per European Organization for Nuclear Research, GC is considered as ''service for sharing computer power and data storage capacity over the Internet'' [1]. Initially, GC was designed to stimulate the scientific computing and research facilities while providing high utilization of geographically distributed sources. Currently, GC provides loosely coupled ''super virtual computers'' to perform ''single'' large tasks that may include DNA sequencing, drug design, protein modelling, prediction of climate change, ecological modelling, gamma radiation detection from intergalactic objects, detecting ''God Particle'' from Large Hadron Collider, and IP telephony etc. [2], [3]. Though, GC is capable to cater complex tasks, it lacks in following aspects, such as: high level common services, individual centric service, data handling, standardization, minimum heterogeneity support, and security [4].

B. CLOUD COMPUTING

In 1997, Chellappa [5] firstever coined the term ''Cloud Computing'' (CC) in his address to the INFORMS Annual Meeting. His idea was more concentrated toward usage of cloud services in terms of economical rationalization among Small and Medium-sized Enterprises (SMEs) or upcoming business models which earlier was never thought of [6]. Existing CC paradigm aims at empowering service providers by facilitating efficient resource management in data centers [7]–[9]. Unlike GC, clouds are capable with the scalability and flexibility metrics. Clouds provide service centric models to meet the user necessities. Applications of CC are now widespread with mailing services to storage, document processing, hosting services, image processing, and video streaming. However, problems associated with current CC are multifold such as: reliability, availability of services and data, security, complexity, limited customization, and cost of service.

C. FOG COMPUTING

It was first introduced by the CISCO corporation [10]. The primary focus of Fog Computing (FC) is to provide ''closer'' computation, data storage and application services than what the CC does. This ''near-proximity'' is related to client-side only, especially on smart phones and embedded systems. Most of the time, FC is incorporated into the local network routers to provide user ''on-the-spot'' services. Technically, FC restricts data mobility to some extent, which results into enhancement at location awareness, low latency, system efficiency, and backbone bandwidth savings. Certainly, it improves data availability and security issues which are the most prominent challenges in CC. Though, FC holds a strong competition in market place, it needs to solve following issues that includes standardization, software package portability among various embedded computing specifics, container management for resource constraint embedded systems, and stringent support system, to mitigate with frequent communications with cloud [11].

D. EDGE COMPUTING

It is the latest upgradation in computing paradigm phenomena that performs data processing at the ''edge'' of the network [12]. Unlink FC, Edge Computing (EC) provides more efficient solution in terms of computation and data-analysis performance at the edge of the network. EC is more inclined toward serving applications that include Internet of Things (IoT). EC may reside in and around of FC, while supporting the backbone cloud service by means of ''Push/Pull'' attribute. Though, several application scenarios are proposed where EC performs better than other computing paradigms [13]–[16], it suffers from problems such as: programmability, naming, standardization, data abstraction, service management, battery life, and cost to end-user.

Till now, the discussions provide comprehensive awareness about various advantages and drawbacks of all the four paradigms as mentioned above. GC supports one task performance in distributed environment, CC provides service models to perform a task, FC allows ''on-the-spot'' on-network processing of task, and EC leverages computation of tasks originated at edges of network. DC goes beyond the existing generic concept of service/network/storage to a sub-platform schema, as presented in the following sections. Table 1 presents the key terms and their abbreviations used in this article.

III. DEFINITION OF DEW COMPUTING

Before going into the detailed discussions on DC, let's first get acquainted with the answer of a vital question: What

TABLE 1. Key abbreviations and their full forms.

Abbreviation	Full forms
AI	Artificial Intelligence
ATAHA	Any Time Any How Access
BD	Big Data
BTS	Base Transceiver Station
CC	Cloud Computing
CERN	Organization for Nuclear Research
DC	Dew Computing
DDNR	Dew Domain Name Redirection
DDNS	Dew Domain Naming System
DHPIS	Distributed High Productive Information Processing
DISE	Distributed Information Processing Environment
DoT	Dew of Things
EC	Edge Computing
FC	Fog Computing
GC	Grid Computing
GISE	Global Information Processing Environment
HPC	High Performance Computing
IaaS	Infrastructure-as-a-Servcie
IaD	Infrastructure-as-a-Dew
ICT	Information and Communication Technology
IoT	Internet of Things
JARVIS	Just A Rather Very Intelligent System
LAN	Local Area Network
LPIP	Low Performance Information Processing
OSI	Open Systems Interconnect
PIC	Personal Information Center
SaadP	Software-as-a-Dew Product
SaadS	Software-as-a-Dew Servcie
SaaP	Software-as-a-Product
SaaS	Software-as-a-Servcie
SME	Small and Medium-sized Enterprise
SSD	Solid-State Drive
TB	Tera Byte
WoS	Web of Science
ZB	Zeta Byte

is DC? In reply, Wang [17] emphasized on two key words: (1) ''independence'' and (2) ''collaboration'' which are to be necessitated for the evolvement of the DC. K. Skala et al. on the other hand, provided a theory on micro-service components with respect to existing centralized service systems. It has also added on how DC and its allied components can successively be positioned far away from current virtual infrastructures. This, apparently refers to a rough-similarity with ''independence'' theory as prescribed by what Wang [18] proposed with a twist of self-sufficient approach. Similarly, Ristov *et al.* [19] prescribed the ''independence'' theory, as of Y. Wang and correlated with the newluy introduced ''collaboration'' theory.

It is comprehended that all the three definitions do not contradict with each other per their souls, however, micro-service and information centric issues need to be added with the Y. Wang's one. Being most novel, DC requires some clear nomenclatures to make it more readable and comprehensive for fellow researchers. Five novel aspects (e.g. Rule-based Data Collection, Synchronization, Scalability, Re-origination, and Any Time Any How Accessibility-ATAHA) are hereby newly included in this definition. Novel service models and identity mapping techniques are also associated for proper conjunctions. Hence, the resultant definition of Dew Computing may be read as below:

''Dew Computing is a programming model for enabling ubiquitous, pervasive, and convenient ready-to-go, plug-in facility empowered personal network that includes Single-Super-Hybrid-Peer P2P communication link. Its main goal is to access a pool of raw data equipped with meta-data that can be rapidly created, edited, stored, and deleted with minimal internetwork management effort (i.e. offline mode). It may be specially tailored for efficient usage, installation, and consumption of local computing (i.e. on-premises) resources like PC, Laptop, Tablet, high end Smart Phone. This computing model is composed of six essential characteristics such as. Rule-based Data Collection, Synchronization, Scalability, Re-origination, Transparency, and Any Time Any How Accessibility; three service models such as Software-as-a-Dew Service, Software-as-a-Dew Product, Infrastructure-as-a-Dew; and two identity models (e.g. Open, Closed). All such efforts shall be made towards running of applications in a purely-distributed and hierarchical manner without requiring continuous intervention from remotely located central communication point e.g. cloud server etc..''

Detailed discussions on all these segments of DC are presented in the following sections.

IV. THEORY AND CONCEPT BEHIND THE DEW COMPUTING

Current section is the core part of this article that seeks the answers to a set of questions such as:

- What are the components of the Dew Computing architecture?
- How is it related to Cloud Computing aspect?
- How do those components work in integrated way?
- What are the features of Dew Computing?
- Is DC feasible to get deployable at current scenario?

Key answers of these queries are placed inside the arguments and state-of-the-art discussions given later. Dew-cloud architecture is the best part to start with. But before that, recapitulation about standard client-server architecture is worth to mention. Client-server architecture is the basis of existing distributed network-application structures. It is designed to partition scheduled/assigned jobs between service provider and possible service seeker. Initially, it was designed to provide distributed service to a client from anywhere in the network. Main goals of this architecture are to leverage four services such as: (1) flexibility, (2) interoperability, (3) scalability, and (4) usability. The relationships between server and client may be formulated by three ways such as: 1:1, 1:N, or M:N, where M, N represent arbitrary but finite number of clients or servers greater than 1, respectively. At bottom line, we may assume that client-server architecture is the ultimate-basis of current cloud computing and internet.

In contrary to client-server, current article proposes a novel dew-cloud architecture.

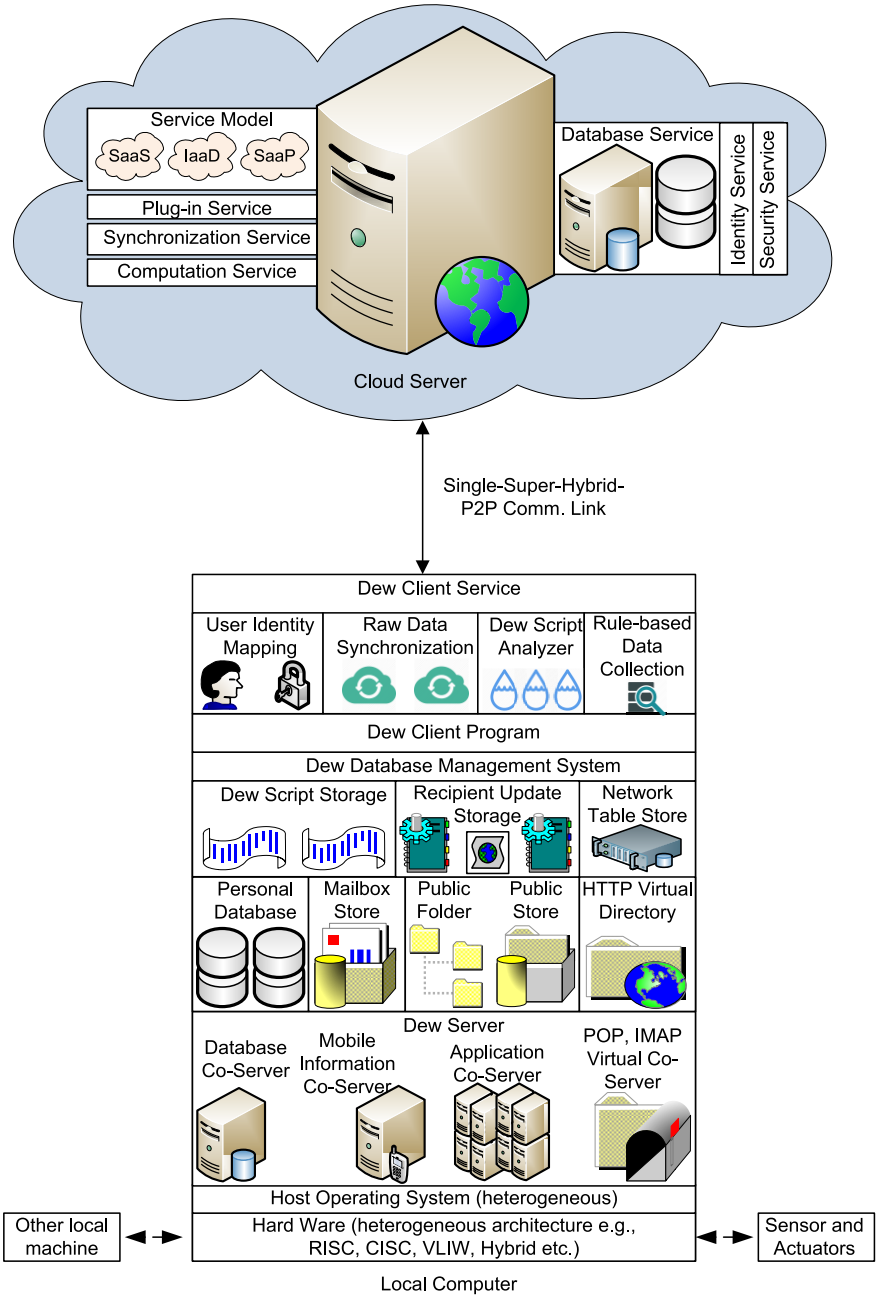

FIGURE 1. Proposed model of Dew-Cloud architecture.

A. DEW-CLOUD ARCHITECTURE

Wang [20] has proposed a cloud-dew architecture to describe what it is and how it works. In that design, Y. Wang assumed cloud-dew architecture as an extension of existing client-server architecture as discussed earlier. Following design considerations were undertaken while illustrating the cloud-dew architecture. For example, server was reconsidered as the "cloud server" that includes all supportive services and database facilities and client was reconsidered as an ordinary "local computer" which comprised of four components e.g. "client program," "dew server," "DBMS," and "databases."

Here, I append few more important and essential parts into it that may appear as a more refined version (see Fig 1) than the earlier. While elaborating proposed architecture, I adhere to some portions of what W. Yang proposed. It is done due to provide more readability and linked-comprehension. In this case, a Dew Server is assumed as a special-purpose web server that inhibits inside user's own local machine. This local machine is comprised of four main components such as:

(1) Dew Server, (2) Dew DBMS, (3) Dew Client program, and (4) Dew Client Service Application. An ordinary cloud is hereby considered as a Cloud Server that resides in a remote location. Main purpose of proposed Dew Server are (1) to server user with requested services i.e. in case of cloud and (2) to perform synchronization and correlate between local data and remote data. Discussions on these components are presented below.

B. DEW SERVER

Dew Server has 8 vital specifications as given below:

i. It must be light-weight i.e. structure, purpose, and usage should be specific.

ii. It will ordinarily serve only one client at a time.

iii. It should normally store the frequently used user data in its databases.

iv. Size of stored data shall be as minimum as possible.

v. It may be prone to data-loss due to its un-regularized interaction with fatal errors like attach by malware, virus or system damages.

vi. Its lost data should be recoverable (i.e. regeneration) from Cloud server.

vii. It will be responsible for all activities, including controlling of DBMS, identity mapping service, raw data synchronization, rule based data collection, dew script analysis etc. Ultimately, it will emerge as a Personal Information Center to its user.

viii. Its services shall be accessible to user at any time with/without internet connection.

Let's compare these characteristics metaphorically. Here, Dew Server is assumed as a dew upon a plant-leaf and Cloud Server is considered as floating clouds in the sky. Dew is less-weighed (as is i), it is placed on an entity (as is ii), dew is collection of sub-molecular ''dewlets'' that captures the surroundings to its center (as is iii), dew is physically tiny (as is iv), dew is evaporable (as is v), dew may be recreated from showers from cloud (as is vi), dew is closest to earth to help wet earth, grow crops, create moisture etc. (as is vii), and it very near to us to touch and feel (as is viii).

Dew Server is the most vital part of the proposed Dew-Cloud architecture. Thus, it must be equipped with several adaptive technologies. Ordinarily, it comprises of four essential co-servers such as: Database, Mobile Information, Application, and POP/IMAP message protocols.

In general, when a user searches or accesses for some website contents in internet, the ''remote-cloud/server'' provides all the responses against such requests. In proposed theory, the user is able to surf the same contents by means of the Dew Server. Presence or absence of internet will not be an issue. For example, suppose there is a website *www.abc.com* which is already adopted with Dew-Cloud architecture and avaibale for being surfed. The Dew Server (residing at user's local computer) must keep note of it and creates (upon confirmation from user or self-motivated) a dual (similar copy) of the visited website in user's local computer. Such type of dual may be called as ''Dew Site'' that is analogous to existing web site. Once such Dew Site exists in the local machine, user can surf the web-information about the Dew Site and performs necessary modifications, if any. But, modifying a Dew Site is not an easy task. A special scripting file coded by a web language may be presented to handle such activities. It is worth to note that each Dew Site should be associated with unique Dew Script counterpart. Now, whatever information is already present in a Dew Site, Dew Script is able to modify it per its supervisor i.e. ''Dew Analyzer.'' A Dew Analyzer is a tool which is designed to take over all possible charges performed by the Dew Script. Hence, a Dew Database should be associated with storing such activities. Database related tasks are normally undertaken by Database Co-Server. For this reason, the Dew Script technology should not be proprietorial to any vendor. Instead it should be based on publicly available and popular open-source technology.

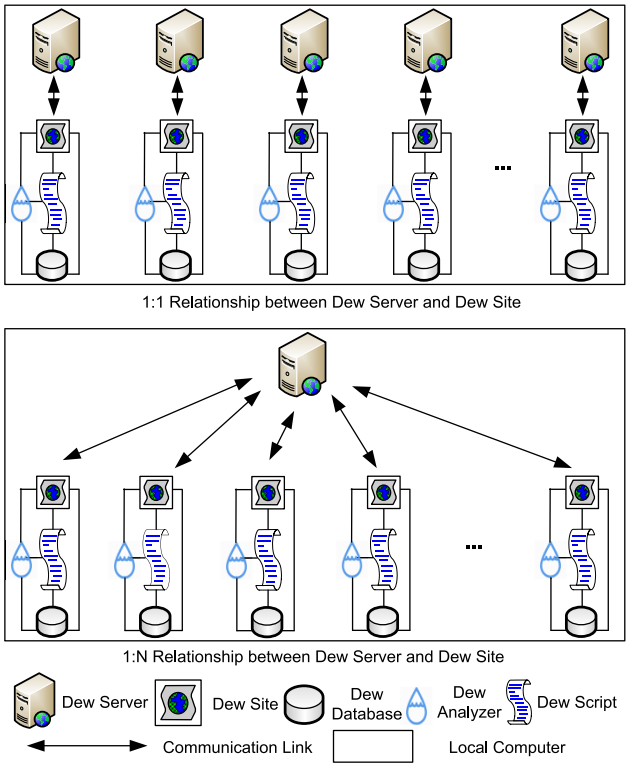

FIGURE 2. Proposed relationship between Dew Server and Dew Site (1:1 and 1:N).

C. DEW SERVER-DEW SITE CORRELATION

Till now, discussions are mainly focused on performance-specific aspects of Dew Server and its allied sub-components. However, it is yet not decided that what will be the number of Dew Servers that will be installed in a local machine. This can be illustrated by the proposed dimensions in Fig 2. Each of the Dew Sites should have its own Dew Database, Dew Analyzer, and Dew Script associated with it. It is possible to have several orientations in this regard in

following fashions: (1) One Dew Server for each Dew Site (i.e. one Dew Script) i.e. 1:1 and (2) One Dew Server for multiple Dew Site i.e. 1: N or in any hybrid fashion. It is also possible for one Dew Server to host an optimized number of Dew Sites. It should be done according to implications of load balancing techniques. For example, the mapping of one or multiple Dew Server versus one or multiple Dew Sites could be done as conjunction to the cache memory mapping technique. In 1:1 scenario, one Dew Site will ordinaily be uniquely handled by one Dew Server. It will resul into a better performing and more secure system. But, it will incur higher cost during installation in terms of service charge, licensing, over burdening, and complexity. It may be a cumbersome situation for local machine to handle too many Dew Servers into one local system. Hence, the local machine should be equipped with multi-core processing units with real-time and optimized scheduling tool.

Otherwise, one Dew Server may allocate multiple Dew Sites in the local machine. This is less cumbersome than the 1:1 approach. However, when dynamic selection of Dew Sites is imposed upon one Dew Server, a specialized scheduling algorithm needs to be developed along with it. Whenever, a Dew Site is not being surfed or idle, Dew Server may temporarily select other Dew Site which is currently over-loaded or its allocator (i.e. Dew Server) is over-burdened. For this reason, a Dew Server should adopt ''Plug-in'' mechanism. Once, a Dew Site is temporarily detached from hosting facility and others are attached, the request coming from user must be channelized. This would require an intra-domain mapping system so that URLs are uniquely provisioned and mapped. It is a major concern for most of the existing scripting techniques and databases offered by different vendors.

Now-a-days, a dozen of Integrated Development Environments (IDE), web languages, scripting tools, DBMSs etc., are present in public domain. Hence, it is mostly impossible for a Dew Server to integrate each and every such platform and technology into its ecosystem. However, if a Dew Server adopts following considerations, it is envisaged that some of the problems might get lowered down such as:

- Removing restrictions on usage of homogeneous technology and platform in Web Site and its corresponding Dew Site.
- Seeking current server packages that leverage feasible supports toward employing multiple DBMS and platforms in the same machine.
- Facilitating a standard guideline on how a Dew Server developer should be cooperative with Dew Site developers.

D. DEW CLIENT SERVICES

Dew Client program is responsible for facilitation of several applications such as: identity mapping, rule-based data collection, raw data synchronization, scalability, ATAHA etc. Out of these, identity mapping may be considered as most vital task to follow which is prescribed as below.

1) SERVER IDENTITY MAPPING

Websites normally check user identity before giving access to secure contents. Login-logout combo is the basis of it. In case of Dew Server, similar approach may be taken. Dew site User Identity (i.e. DUI) could be a possible solution. However, if one user access one Dew Server (i.e. 1:1) then such identification may not be necessary. Because, the user herself has access over own dew computer. But, if multiple users try to access one Dew Server at a time, then it becomes a necessity. Otherwise, a gross privacy violations may occur. DUI may be based on existing privacy and authorization protocols e.g. login/logout using some user id and password, as in case of standard website user identity i.e. WUI.

2) USER IDENTITY MAPPING

It is obvious that the correlation between the WUI and DUI is a trivial task. To tackle this probem, a complying solution is proposed in form of the user's activity diagram as shown in Fig 3. Let's illustrate a use case scenario as follows. Suppose, ''*keshav*'' is a local user i.e. Dew User and ''*partha*'' is a global user i.e. ordinary internet user. Both the users want to access *www.abc.com* (assumed as an ordinary Web Site) which is hosted on a cloud server and imported Dew-Cloud architecture. Assume that, any naïve user who wishes to access and modify selective contents in the *www.abc.com,* can do so, provided internet connection is available. For sake of minimizing overhead in processing, authenticating, and hassle-free service,weI compel all users to have individual OpenID. It is an open standard as well as a distributed protocol, meant to provide authentication services to any user who already owns such user ID from any certified OpenID provider and uses that user ID to login any Web Site which accepts OpenID's authentication service, to login *www.abc.com*. Now suppose, *keshav* (i.e. *keshav.google.com*) and *partha* (i.e. *partha.google.com*) both of them request and get individual OpenID accounts (see Step 1 and 2 in Fig 3). In this context, both of them can access and modify their own data (READ, WRITE and EXECUTE mode but other's data in READ mode only) *www.abc.com* (see Step 3 and 12 in Fig 3). Such access is analogous to existing web sites e.g. *www.facebook.com,* www.twitter.com*, and www.linkedin.com* etc. When the internet connection is available, both of them can access on the Web Site and everything seems to be fine enough. Simultaneously, when *keshav* does his first-time registration and login to the Dew Server. He accesses and modifies (if any) personal data which is otherwise visible to any internet user on *www.abc.com* (see Step 4 Fig 3). Upon successful browsing of *www.abc.com,* he wishes to download some personal data (i.e. to be stored in Dew Database) and profile-page to the Dew Server (see Step 5 Fig 3). This data collection could be visualized as per user's prompt or predefined rule-based. After completion of Step 5, Dew Server will create a Dew Site in name of *wid.abc.com* in local machine. Here, *wid* is incorporated in analogous to *www* to differentiate between

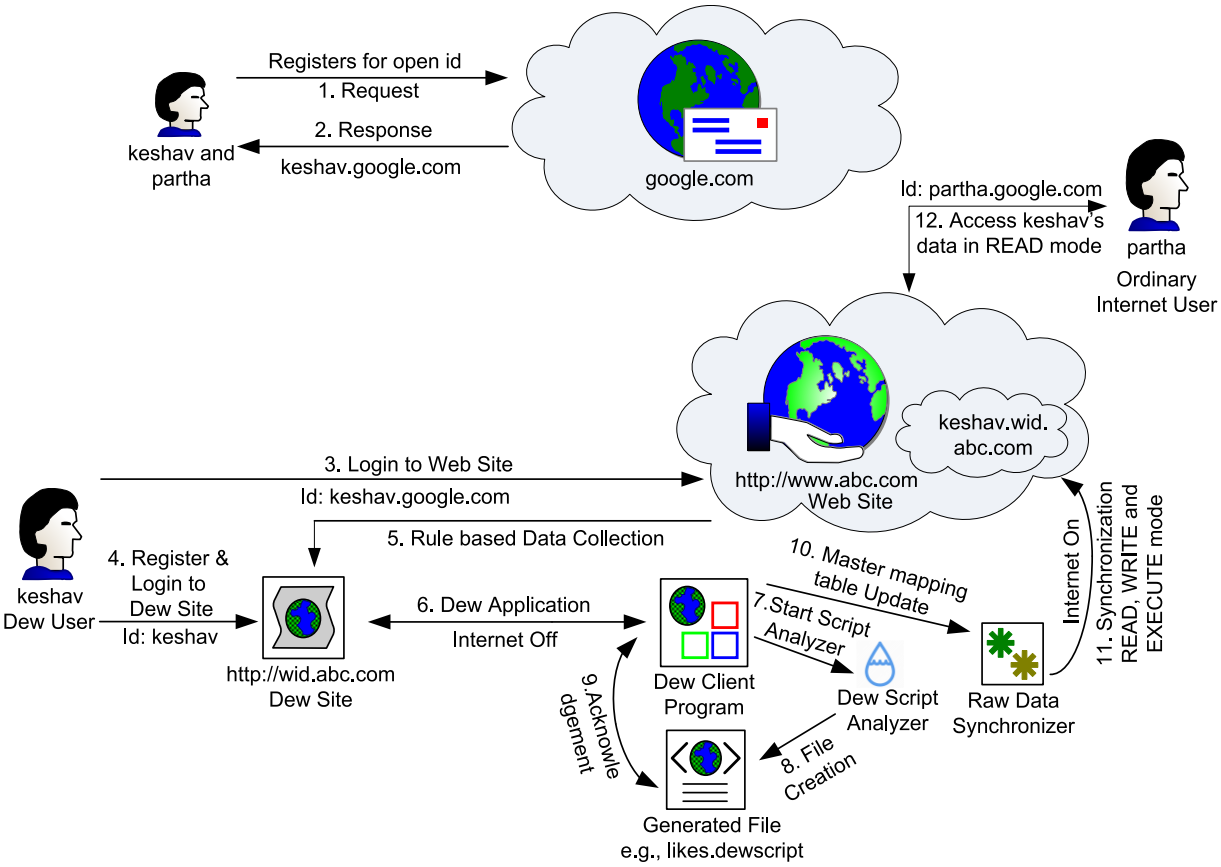

FIGURE 3. Activity diagram while surfing on Dew Server.

Dew Site and Web Site. This Dew Site provides *keshav*'s personalized view of *www.abc.com*.

Now suppose, the internet connection is not available to both the users. Thus, *partha* is not able to access *www.abc.com* although his data are currently online. He should wait till the reconnection with the internet. But, *keshav* now has an option. He may surf his own information and post any message and write on some links in *wid.abc.com* (see Steps 6, 7, 8, and 9 in Fig 3). During such activities, Dew Client program starts services of Dew Script analyzer which in turn coordinates in creating new file such as *likes.dewscript* or appending information to existing Dew Site-pages. When internet is again connected (later), updated-information on Dew Site-pages and already created files are synchronized with *keshav.dew.abc.com* (partial data storage facility of *www.abc.com*). It is a private memory location for all information about *keshav's* profile. Any internet user (like *partha*) is now able to surf *keshav's* profile on *www.abc.com*.

But, there is a problem i.e. ownership of updated files. How the synchronizer would identify the owner of *likes.dewscript*? As we know that the owner of *likes.dewscript* is *keshav* not *keshav.google.com.* So, when *keshav* would try to reuse (re-update) or surf this file later the *www.abc.com*, related cloud services will not authorize i.e. allow *keshav* to do so. This problem may be solved by incorporating a master mapping table into Dew Server. It will periodically map Dew User's identity with the cloud user. This copy may also be shadowed and the shadow part may be placed in the cloud server.

E. DEW DOMAIN NAME SYSTEM (DDNS)

If one Dew Server hosts one Dew Site, it would be easier for the system to handle all the requests coming from user to direct. It is very likely that one Dew Server will host multiple number of Dew Sites in practical scenarios. In that case, a redirection strategy is a must for the dew computer. To resolve this problem, we introduce the Dew Domain Mapping System i.e. DDNS a modified version of existing Domain Naming System. It is well known that a global DNS maps a requested ''string'' into an IP address. DDNS does the same thing, but with an exception that the redirection of local URLs are done with respective to the target Dew Sites. It can be done by two ways as precribed below.

1) METHOD 1 (HOST-TO-IP MAPPING)

In this method, local URLs are mapped to a standard localhost i.e. 127.0.0.1. Here, local URL means URL of individual Dew Sites e.g. *wid.abc.com*. Instead of sending local URLs to the DNS service, Dew Server sends those to DDNS. Mapping of

local URLs could be done in three ways such as: (1) *zone file:* it contains mapping information between domain names and IP addresses, usually organized in form of text representations of resource records. The domain owner should add specific records into the domain's register. Obviously, without internet connectivity such file generally impossible to get updated in cloud side. The cloud server must incorporate Dew-Cloud architecture to get simultaneously involved, (2) *hosts file:* it is a plain text file, placed in every operating system to map hostnames to IP address. All local URLs could be mapped to localhost e.g. *wid.abc1.com, wid.abc2.com, . . . , wid.abc10.com* to *localhost-127.0.0.1*. This method could be useful for any machine that is independent of Dew-Cloud architecture, and (3) *development of novel web-browser:* existing web-browsers should be modified to co-opt with surfing of Dew Sites. Whenever, the browser will be opened, it should be capable to map the local URLs into localhost.

2) METHOD 2 (DEW DOMAIN NAME REDIRECTION-DDNR)

In this method, Dew Server redirects all local URLs to the requested Dew Sites only. For example, if a few particular requests about *wid.abc1.com* and *wid.abc2.com* arise from a dew user, those requests must be mapped with *wid.abc1.com* and *wid.abc2.com*, respectively. This task is plausible because a portion of the host name is usually captured by local environment variables. Hence, whenever any URL request comes from th dew user, the environment variable would assist the Dew Server to redirect into the specific Dew Script.

F. TRANSPARENCY

Transparency refers to the process of separation of the high-level abstracts from low-level implications [21], [22]. A system that "hides" the complex implementation details from its users is called a transparent system. Oszu and Valduriez [21] mention two forms of transparency such as: data replication transparency and data distribution transparency. In the context of DC, both of the forms may be leveraged [23].

Data replication transparency refers to the user's accessibility over data as if it were a single copy, though the data may have two copies-one stored in the Dew Database (i.e. local copy) and the other into the cloud database (i.e. master copy). User may or may not be aware of such duplicacy.

Data distribution transparency refers to the fact that users shall not have any difficulty to handle to data whether it may be stored in the on-premises computer or the cloud database. Though it should, data replication and network transparency are not always perceived in existing DBMS and other system solutions. If it were, the scenario would have been as follows: suppose a user is currently accessing *www.instagram.com*. If the internet connection is off, then this web site is also not accessible. Now, if user wants to access *www.instragram.com* in current context, replication and network transparency should play an important role. This means, when internet connection is on, on-premises computer i.e. Dew Server will store the visited page and store all information in local database i.e. Dew Database. Upon suitable modification in the behavior of the browser, user could become able to access *www.instragram.com* in local machine. The browser will first try to connect the internet, if successful, the surfing would continue as usual. If there is no internet connection, the Dew Server shall leverage similar facilities that would have been if it were provding the original web surfing experience. Here, Dew Server may allot all the replicated data to the users, while hiding all the complex details about communication link failure, existence of operation from local computer, and browser behavior. The user will assume as if he/she is viewing the pages of *www.instragram.com* on internet itself.

Will this transparency provide exact services as the original web site would have given? Due to the illustrative formation of the Dew Server, following activities would be possibile such as: real-time image posting, message posting, and poking. But, it may not be "that much" realistic. Perhaps, a more drastic approach is to be comprehended. It is better to keep the user aware of current context i.e. what is going inside the system. In that case, a *mutual consistency algorithm* needs to be devised which will synchronize the replicas in both Dew Database and cloud database upon successful internet connection.

Mutual consistency algorithm may incorporate two orthogonal approaches to solve this problem. Firstly, high-priority update and low priority update; high priority update takes place in context of global access to cloud server; whereas low priority update takes place after high priority updates have already occurred. Secondly, core update propagation and distributed update propagation; core update necessitates that data update to be first taken in the *master copy*; and distributed update implies on *local copy* after *master copy* is updated. Hence, four relationships between these two update strategies could be framed: (1) high core priority, (2) high distributed priority, (3) low core priority, and (4) low distributed priority.

In applications, it may be at loss due to its less affection toward the internet connection. In this situation, high priority data replication will neither be possible nor appropriate. Thus, leaving a choice for low priority update. Now, out of the two options i.e. low core priority and low distributed priority, low core update seems unpredictable due to frequent internet connection loss. It leaves a last possible option-low distributed data replication. In this strategy, *local copy* gets asynchronously updated sometimes after the original modifications are processed. Obviously, such updates shall leverage more complex communication and database protocols.

In reality, Dew Database presents a partial replica of the cloud database, thus a forming a subset of cloud database. The cloud database shall contain a complete set of data about the user and its web site access. Whereas, the Dew Database stores the local data-logs only. There may be a situation when the Dew Database would require oweing the whole dataset or try to contain some amount of data only at

Dew Database. Undoubtedly, such incidents may occur when (1) the data is completely related to Dew Server's operation and (2) the partial amount of data is very secure to user that he/she does not want to replicate in cloud.

G. SINGLE-SUPER-HYBRID-PEER P2P NETWORK

As already discussed, proposed dew-cloud architecture is an extended form of existing client-server architecture. Hence, it could be assumed that a peer-to-peer (i.e. P2P network) communication system is inherited inside the dew eco-system [24]–[27]. But, theres is a steady exception. Dew-cloud architecture requires servers at both ends (i.e. web server end and client end). Thus, the network structure of dew-cloud architecture may be perceived as a hybrid P2P network instead of a simple P2P network. Here, the cloud server acts like a super-peer node. At any time-stamp, one Dew Server communicates with one cloud server. Thus, envisaged P2P network structure could also be reframed as a Single-Super-Hybrid-Peer P2P network. If this communication link (i.e. between the super-peer and a normal peer) fails, selected peer is partitioned among whole dew-cloud system. In such scenario, the client-peer is requested to perform several tasks so that the user may be facilitated with uninterrupted web surfing. By this virtue, a local peer not only communicates with super-peer, but also talks with similar client-peers. Thus resulating into a new opportunity to create a closed communication group among multiple local-peers without intervention from super-peer.

It is henceforth easy to understand that a Single-Super-Hybrid-Peer P2P network is simpler than multiple-super-peer P2P network but more complex to implement than a client-server architecture.

H. SERVICE MODELS

Software-a-a-Service (i.e. SaaS) in one of the most vital service models in the CC paradigm. In a SaaS model, a software is usually placed over a centralized cloud server and accessed by multiple users in subscription basis. The whole task of such access is usually leveraged by a web browser or with licensing clauses. Before origination of the SaaS model, a similar but less popular business model existed in the market i.e. Software-as-a-Product (SaaP) [28]–[31]. In a SaaP service model, an end user hosts a purchased software from designated server and use it in a local machine. Licensing becomes mandatory for such activity. Though, SaaP existed earlier than SaaS, few organizations do still provide similar services, for example, Microsoft Office. SaaP does not require instantaneous internet connectivity to server its jobs, but the user should perform installation, medication, and update manually. It is a fact that, SaaS is somewhat better than SaaP in multitude forms that includes (1) centralized data storage, (2) automatic data update, and (3) free maintenance. A report says that SaaS has ''better economic model'' in terms of low cost and auto update than SaaP [28]. Despite of SaaS's service-ability, it lacks in one issue that is internet connectivity. For example, suppose a shop owner owns an account in the Salesforce to manage all the store related information and local power goes off. The owner will certainly be unable to update the real-time store information to Salesforce and business will be running manually. Similarly, when a shop owner owns a business where internet connection is very infrequent or completely nil. He may not be able to access SaaS service at all. Instead, he should use SaaP to run his business transactions. This implies that both the SaaS and SaaP models have advantages and drawbacks.

What if we could merge these two? But how? One solution is to simultaneously facilitate both of the SaaS and SaaP metrics to a dew user. When one goes offline, other holds the business. But again, there is a problem-some data will be stored in cloud server (i.e. SaaS) and some will be in local PC. Now when the user is unable to access both of the storage locations what will happen? In Dew Computing, such solution may be provided. As already discussed, in DC environment, a user has no obligations toward mandatory usage of internet for remote data access. Hence, user can access and update the information stored in local machine. Whenever internet connection is established the updated information may get synchronized with the cloud server. Software-as-a-Dew Service (i.e. SaaDS) and Software-as-Dew Product (i.e. SaaDP) are such two notions by which local and distributed software access could be resolved.

Infrastructure-as-a-Service (i.e. IaaS) is another important cloud service that provides virtualized computing resources to users. User, only needs to login the cloud platform and relevant services will be in place. But in DC, cuurent perspective is completely different. Here, user gets complete access to the on-premises machine without internet accessibility. This provides a good opportunity for this service to be reframed as *Infrastructure-as-a-Dew Service (i.e. IaadS)*.

Is IaadS just an alias of IaaS? Certainly Not. Because, the local dew machine as got the real-time and dynamic support from the remote cloud server. If data or login information is lost or the local machine is stolen or destroyed, data will be safely recovered from the cloud. How? On-premises computer must store all settings related data as a backup copy into the cloud. Upon any damage to local copy the same may be retrieved from cloud and be placed in a newly bought machine. This scenario resembles to the ''regeneration'' metaphor of dew drop from the clouds.

I. SYNCHRONIZATION

Synchronization refers to the maintenance of two or more copies of data that are placed in distributed locations in a way such that seamless coherence among the multiple copies exist and data integrity should retain. Usually, this process needs to be synchronized with multiple sets of data in correlation with each other. In process synchronization, multiple distributed processes join to accomplish a common task sequence. In dew-cloud architecture, synchronization makes a clear presence in terms of applicability, importance, and sustainability. For example, suppose a user wants to access www.gmail.com/drive. The user must first authenticate the

''user identity'' with this web site. Similarly, the user must authenticate ''user identity'' for the Dew Site. When both of the authentication processes are over, a user-defined synchronization tool would upload/download the user data from/to Dew Server or cloud server. Certainly, this synchronization process shall be primarily based on time-stamps on each of the data-files.

V. APPLICATION POTENTIALS

This section presents key impacts of dew-cloud architecture in multitude forms. Application potentials are hereby investigated to prescribe possible application area of DC. A novel 4-tier architecture follows the discussion that creates a symbiosis among cloud, fog, edge and itself for providing better user experience towsrd web surfing.

A. IMPACTS OF DEW-CLOUD ARCHITECTURE

It would be wrong to interpret that Dew-Cloud architecture will automatically come into the scene of play. Major modifications and novel integrations are hereby needed to assimilate with rectified conjuncture. It is envisaged that when an enterprise or a cloud service provider would adopt the Dew-Cloud architecture into their ecosystem, the cloud will be capable to cooperate with many Dew Servers, simultaneously. Clouds shall be responsible for world-wide service providers while Dew Server should leverage direct data access to its user. Several application specifics will be ordered, as prescribed below:

- A user is capable to store and retrieve data to and from CC services as usual. More importantly, the user is also empowered to immediate data transceive with the Dew Server, even when the internet is not available. This gives user a unique opportunity to incorporate Any Time Any How Access (i.e. ATAHA) paradigm.
- Till date, web surfing without internet is an impossible job. Presence of the Dew Server would provide related as well as qualitative advantages. Now, the user can surf websites in ATAHA manner, despite of the fact that the current data should be pre-loaded in the Dew Server by rule-based data collection engine. Whenever, the internet connection is established, the processed data will be seamlessly synchronized to the designated Cloud Server. Obviously, a Dew Server cannot accommodate enormous amount of data which are part of cloud. Only a finite amount of most frequent and selective data to be provisioned in Dew Server.
- User is now capable to access any software using SaadS or SaadP model. Both of the cases, the software service services will be provided in form of the web sites (as well as the Dew Site). Now software making process could be unified by assimilating a Unified User Interface (i.e. UUI). Now, software learning and training will be much simpler than earlier.
- Another aspect of DC is the flexibility of software cost model. Upon a common agreement between Dew User and software vendor, license of the software may be provided to the Dew User. But, now Dew Users need not pay the whole amount to the vendor. The process of purchase, licensing, and usage could be logged onto the cloud database. Whenever, Dew User authenticates him/her self with cloud, he/she will be allowed to access the software in local machine and the cost may be deducted from bank account in subscription basis. Such type of business model may help SMEs to grow and leverage software services in SaaP (i.e. Dew Site access) model too.
- The principal aim of cloud or fog computing is to address complex computational operations which are regularly performed on bulk amount of data. Those data usually come from some web base or system level activities either performed by an end-user or system administrator around the globe. Most of the time, those data are in raw format having no context-ful information (i.e. context-free). In case of DC, all raw data should be tagged as context-aware. It means that the data would be accompanied by meta data. Having control over context-aware dataset that the DC would leverage for all types of self-organization in system level. Thus, providing more capability toward in-depth analysis and knowledge discovery. Despite of complex data processing capability, CC and FC are ordinarily not tailored to cope up with context-discovery solutions, that results into a gap of plausible generic integration of context-free data in data mining notions.
- The goal of DC is to fully realize user's need and its environmental interactions in correlation with CC and FC. To solve this, Dew devices need to seamlessly cooperate with each other. The reasons foc such behavior are low computation processing capability and less affection toward internet usage. Hence, Dew-Computers should act like self-organisms which are very similar as the living examples around our environment. Being space-oriented, Dew-devices are expected not getting connected with internet in most of its operating durations. Hence, they should collaborate with other Dew-devices in near vicinity by means of short-range communications.
- As opposed to CC and FC, DC is less resource hungry. Resources in form of processor, memory, and parallelism may create obligations to its normal activities. Thus, Dew-devices should make a nexus among other Dew-devices in cooperative manner where every peer should contribute some portion of its resource for execution of given task. Though, Dew-devices could be comprehended as Low Performance Information Processing (i.e. LPIP) systems, it could provide higher computation capacity and greater decisions, when successful self-organism systems are in place.

B. APPLICATION POTENTIALS OF DEW COMPUTING

Dew Computing paradigm is envisaged to enhance current pervasive implementations in more flexible and

distributed fashion. The area of such implementations is unlimited, that includes home automation, smart gardening, green computation, smart health care, traffic signal monitoring, context-aware communication, smart wearable computing, and industrial gas leakage monitoring. Out of all, IoT could play a major application role along with DC. As discussed in Section I, IoT is an application perspective model where heterogeneous devices act together to perform a set of tasks. Most of the time, aggregated data and inherent information are stored and extracted (respectively) in the allied cloud services. It means that all the ''edge-devices'' in such application scenario need to correlate with internet availability all the time. But, it may not always be viable to use internet due to some reasons (e.g. trivial aspects of local environment, socio-political issues, cost, disasters and behavior of user etc.). Hence, it would be better if IoT could be juxtaposed with DC in certain cooperative manner. I call this new phenomenon as Dew of Things (i.e. DoT). Here, I replace the term ''Internet'' by ''Dew'' as Dew-device shall provide all necessary supportd to the ''things.'' Obviously, when required, ''things'' will communicate with the cloud for system update and fetching necessary information. For example, suppose user-A employs an auto sprinkler system in his/her garden that is integrated with soil moisture sensor, humidity sensor, temperature sensor, digital valve, and a cloud service. Based on the value of soil moisture level, digital valve will automatically replenish some amount of water in the garden. Correlation between air temperature, humidity, and soil moisture level is a thus complex task. If cloud services were employed, it would have received timely values of temperature, humidity and soil moisture and store into the cloud database. Upon an occurance of high drought situation in garden soil, cloud services would have pinged user-A (by email, SMS or push notification) to take necessary steps. When the digital valve would not have been able to communicate with the internet, then it would certainly be unable to sprinkle water. Applying Dew Server, may solve such problem due to the fact that information is mostly locally processed and stored. The control over grading system seems to have more grip in the hands of user-A. Dew Server could efficiently allot and ping all the installed sensors and actuators in garden and provide full network access to all these devices.

As in DoT, other smart computing systems could easily be developed in and around the DC environment. Whatever the application may be, DC is meant to uplift green computation to a new horizon. In normal scenario, a Base Transceiver Station (i.e. BTS) performs all telecommunication realted facilities that include call send, call received, location monitoring of the phone user, SMS service, IP telephony, and video data transmission etc. On an average, a cell tower consumes 3-4 KW power to serve communication services. The AC backup generators generate about 40-70 KW power to support a cell tower. These power generation services are continuously leveraged for normal operation. Assuming DC in operational mode, users will surf web sites in Dew-devices without using internet. Thus, reducing the data traffic on inter-network which in turn reduces work load of cell towers i.e. cell towers will just concentrate on basic voice or text services rather on all superficial services. This scenario implies that cell towers will be normally under-utilized than earlier. Thus, cell towers will perform less computations that will less power consumption. Hence, green communication aspect is possible to be established. When, Dew-devices are sufficient enough to allow communication among each other by means of local network, cell towers will lesser utilized. Thus, resulting into more energy efficient eco-system which does flagship of the concept of green computing. Green computing can be more ascertained with proper understanding of overall interactions between cloud, fog, edge and dew, altogether.

To facilitate such problem, a novel 4-tier dew hierarchy is hence presented in Fig. 4. This hierarchy consists of 4-tiers of computing paradigms. Tier 1 is hosted by cloud specific structures. Tier 1 is adjacent to Tier 2 that comprises of fog nodes and related services. Similarly, Tier 3 is made of edge computing implementations whereas Tier 4 presents dew computing notions. Each tier provides specific and relevant services to its adjacent tier. All tiers except tier 4 have associated with respective firewall services. That inherently improves overall security concern linked to this structure. Dew computing acts at the bottom most layer by paving appropriate synergies and synchronized behavior to all other above tiers. Virtual dew cluster is sldo envisaged that can provide purely ad-hoc service distribution and web-data sharing among inter-dew computers. The communication links between every tier is dependent on individual protocol layer. However, a dew computer may fetch required amount of web-data from edge, fog or cloud counterpart. Similarly, web-data synchronization is also based on inter-alia behavior. Dew server has complete autonomy in retrieving and syncing own data from any source that may be metaphorically represented by existing memory hierarchy in digital systems. It is worth to note that dew server makes its user's experience super flexible by automating the choice of data correlation with edge, fog or cloud services. When requested web-data is not available in dew database, edge database will be searched for. Upon request failure, fog database will be sought and lastly cloud will server the purpose. In Fig. 1., direct connectivity between dew and cloud is shown. This tier replicates the same with an extension of intermediary layers of edge and fog based services. Thus, it may be inferred that dew server is mainly dependent on cloud for web-data search, retrieval, and synchronizing activities. In other aspects, proposed tier structure does fine job in leveraging user centric web surfing experience in better form. User's request is now being fully served without or minimal usage of internet connectivity.

VI. POSSIBLE CHALLENGES

Being a novel computing paradigm, DC faces several technical challenges that includess power management, processor utility, data storage, viability of existing operating system, network model, communication protocol, programming principles, personal high productivity, database security, and

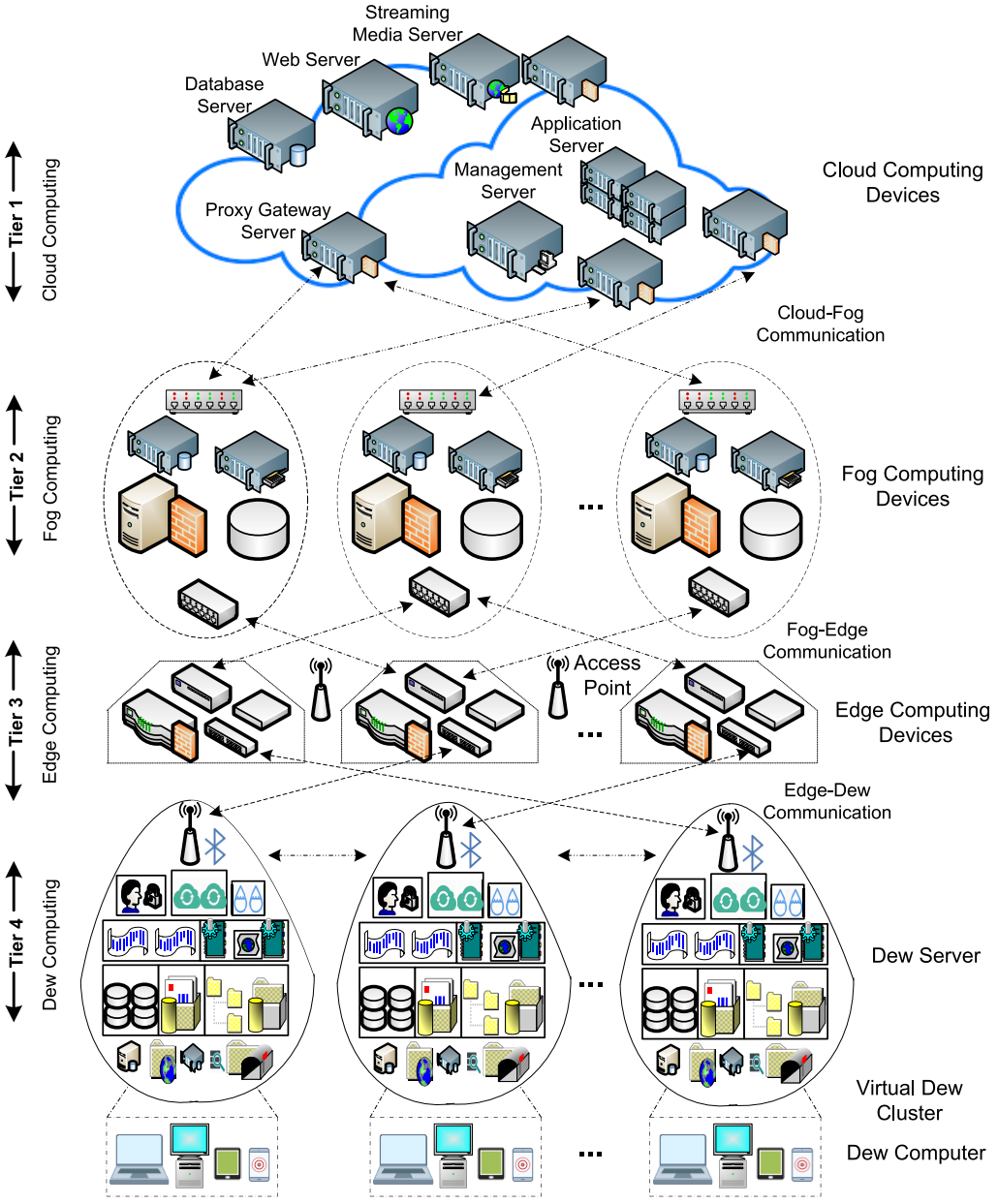

FIGURE 4. 4-Tier Dew hierarchy structure. Tier 1 is positioned by Cloud computing devices and services, Tier 2 comprises of Fog computing devices and services, Tier 3 consists of Edge computing conjuncture, and Tier 4 leverages Dew computing assistance. Each tier is communication-wise directly linked to adjacent tier for efficient servicing to user's request. Dew server is implemeneted on Dew computers that collectively makes virtual dew cluster together.

behavior of browsers. I discuss each of these challenges individually.

A. POWER MANAGEMENT

It is already discussed that DC shall require a local Dew Server to exchange information with cloud in 24×7 manner resulting into a continuous operational environment with or without interaction with both of the user and internet. Henceforth, the on-premises computers shall consume too much power for its activities. To tackle this problem, stand-by mode may be incorporated as it is the case with modern smart phones. Whenever user needs a service Dew Server could be back into normal operation mode otherwise it would stop all front-end activities but the back-end processing might be running.

B. PROCESSOR UTILITY

Now-a-days, most of the computing machines use inbuilt multi-core processors that most of the times are noticed to be extra-gated or supernumerary to the ordinary

user requirement. Dew Computers must enable such scalable mechanism that may upgrade or downgrade the real-time needs of to execute processes. Two solutions are envisioned such as: (1) instead of using high-end multi core processors just employ single core processing units, or (2) in exceptional cases, borrow processing capacity as Infrastructure-as-a-Dew Service from the cloud.

C. DATA STORAGE

This is really a big problem in DC paradigm. Dew machines are expected to belong from smart phone and other smart handheld machines where total memory usage amout is limited. It means an alternative way out should be paved. Several cloud-based solutions are available in market that provide storage facilities, for example Dropbox. It is one of the popular distributed data storage facilities which provides 24×7 service availability to users regardless to its server's availability. It implies that without internet connection, user may upload and download files to or from Dropbox server. Whenever, internet is stabilized the synchronization is done. It leverages a perfect synergy with Dew Computing where independence and collaboration both are employed. Similarly, Microsoft's OneDrive, Google's Google Drive Offline are other storage solutions that is coherent with Dew Computing aspects. Regularly, we are vetted with many applications and services from different vendors on the web. As discussed, several Dew Computing applications are already present in the market but due to lack of knowledge and ignorance, we never tried to understand what these actualy are. Invasion of novel storage techniques are need of the time. Storage in Dew i.e. SiD may be comprehended as a complementary aspect to this problem. SiD, upon realization, could be positioned into on-premises computers that should be supported by current Solid-State Drive technology and integrated storage-software.

D. VIABILITY OF OPERATING SYSTEM

Per usability, operating systems may get incorporated with a set of service-specific modules to facilitate dew applications. Besides process, memory and device management, dew management could be added on top of operating system. In-built dew services could be a core or extended function of such operating system. This would reduce the self-imposed load Dew Server, hence disseminating a better DC services. Another very interesting approach could also be furnished i.e. scalability. Operating system may leverage special module to manage scalability in internal level.

E. NETWORK MODEL

We can assume that each Dew Computing device will collaborate with cloud in its own way, unless those are standardized. On other side, initially very few applications will run on Dew sdevices. But, this number may increase in future. At that time, each dew application shall require its own communication ports to interact with other on-board or distributed resources. What if, these applications cause resource conflicts such as port availability? Let assume that dew operating systems will handle this issue. Then only, redundant codes shall remain inside each of these applications which seems extravagant and unnecessary. Hence, it would be better if new network model be levied upon. On solution is to include one or more layers (e.g. context layer, dew application layer etc.) on top of existing Open Systems Interconnect model.

F. COMMUNICATION PROTOCOL

As already discussed, innovative dew applications are in verge to appear in near-future. New communication protocols are hence indeed a need to cope up with novel hypertext information passing, propagating among peers, and visualization. This would help in autonomic movement of web-information through different OSI layers and be communicated with other similar dew-devices in periphery.

G. PROGRAMMING PRINCIPLES

Revision in network and communication protocols will not be able to persuade dew applications. The reason is simple, current programming models solve how data to be organized and accessed. But, DC paradigm shall require data to be context-aware, exchanged with peers / clouds and processed at very end of Dew Server. Hence, new programming models must be developed to undertake context-awareness and orient synchronization and flexible toward effective way.

H. DEW RECOMMENDER ENGINE

Dew servers must have the capability of providing recommendation on the basis of user's behavior and content surfing history. Novel recommender engine should be designed to enable user to choose/select specific dew sites, thus enhacing the overall user experience. Diverse machine learning algorithms could be tested against development of effective recommender engine. Its main purpose will be to help dew server select/choose another resourceful-dew server from local network for data renting. It will also recommend the dew server when to seek for internet connection and when not.

I. LOCAL DEW NETWORK

As, the DC is designed to be less affectionate toward internet connectivity, it would hevily rely on own dew data and local dew device data. Existing local network protocols may not be sufficient for renting the surfing-data from other dew devices placed in near vicinity. For this reason, Low Power Wide Area (i.e. LPWA) network could be used as complementary support to current network infrastructure. It will reduce data traffic on current network backhaul and increase low-power but moderate spectrum functionalities. IEEE 801.11ah, 802.11af and Bluetooth 5 may also be useful for creation of local dew network.

J. PERSONAL HIGH PRODUCTIVITY

It is important to seek for personal high productivity issues that may be raised in the DC paradigm. As envisioned, Dew-devices shall be less complex and resource intensive

with respect to CC and FC. Hence, there will be high chance that these devices become less productive in terms of information processing capability. As Dew Computers are meant to provide efficient Personal Information Center-wise services, novel information processing solutions are in place. One solution could be to incorporate Distributed High Productive Information Processing notion into its environment. The main goal of such processing should encompass Giga/Tera/Peta/Exaflop computing capability to DC paradigm. It is not very hard to achieve, as DC communication link is assumed to be Single-Super-Hybrid-Peer P2P network that gives the opportunity to assimilate and share resources among local peers.

K. DATABASE SECURITY

As we know that Dew-Server will store a portion of data in local Dew database which upon requirement will be synchronized with cloud database. It is worth to note that both of the data and database are key components of Dew Computing scenario. Despite of high-end flexibility services of current database solutions, one important issue may be missing i.e. security-id database access. We are not talking about the generic login/logout and encryption security that are already present in existing databases. Rather, we are more concerned about the mitigation and migration of cloud-side technologies into the Dew-side. Cloud database is generally supported by the super-specialty security solutions which may be very minimal in case of Dew database. Hence, a threat in Dew database will be much more viable than it would be for cloud-side. Some malicious software or user's fault may bring huge catastrophic error in Dew-system level. It the duty of the dew developers to seriously handle database security from a novel perspective so that hazards in Dew-databases may be efficiently dealt with.

L. BEHAVIOR OF BROWSER

Browser is an application software meant to enable user to search, surf, and perform applications on the web. Novel development in browser technology is a must to manage all the web-related tasks in the DC paradigm. Because, a Dew-Server may host multiple Dew Sites which the user would require while surfing at any point of time. Hence, the browser should be able to relocate the Dew Site, being hosted inside the Dew Server. Obviously, WiD concept might be helpful for the browser to identify and propagate internal user requests to the designated Dew Site. Local Domain Naming System could be another helping tool in this regard.

VII. CONCLUSION

This paper proposes a novel Dew-Cloud architecture which is an extension of existing client-server architecture. Dew is a special purpose server which resides inside user's local computer. It serves user's requests in form of services, as it would in case of cloud. Multiple users are enabled to work in a single Dew Server at any point of time. Dew-Cloud architecture paves a novel computing paradigm i.e. Dew Computing. In purview of Dew-Cloud architecture and understandability, a novel definition is presented. Six key characteristics including rule-based data collection, synchronization, scalability, re-origination, transparency, and any time any how accessibility are discussed. It is envisaged that upon realization of Dew-Cloud architecture, computing scenario shall notably be changed where users are given more flexibility to have control over his/her own web documents and related activities, as is now. The major advantages of proposed DC would be envisaged to be seal healing attributes, autonomic augmentation, self-adaptive, transparent, user-programmability, and extreme scalability. It is expected that due reduction of high internet usage might come into the existence very soon. Being at the bottom most layer of hierarchical structure DC holds the responsibility to simulate the most critical implications of human-digital interaction at the grass root level. It must leverage two key aspects such as self-organization and mutual collaboration in all possible applications. While performing multitude of applications, DC should disseminate behavior of each components in its ecosystem. Certainly, DC would uplift current computing scenario to Global Information Processing Environment, which is a long-cherished need of civilization. But before that, it must cope up with local Distributed Information Processing Environment. More or less, DC will assist in leveraging the propaganda of being greener on its inherited values by minimizing cost, energy consumption, and hassle-free super flexible services.

REFERENCES

[1] CERN. (Dec. 2008). *The Grid Café—The Place for Everybody to Learn About Grid Computing*. Accessed: Jun. 5, 2017. [Online]. Available: http://www.gridcafe.org/
[2] I. T. Foster and C. Kesselman, *The Grid, Blueprint for a New Computing Infrastructure*. San Mateo, CA, USA: Morgan Kaufmann, 1999, accessed: Jul. 7, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=289914
[3] M. Gortz, R. Ackermann, J. Schmitt, and R. Steinmetz, "Context-aware communication services a framework for building enhanced IP telephony services," in *Proc. IEEE ICCCN*, Oct. 2004, pp. 535–540.
[4] G. Allen. *Potential and Challenges of Grid Computing*. Accessed: Jul. 2, 2017. [Online]. Available: https://www.cct.lsu.edu/~gallen/Presentations/MardiGras_GEA_Jan2008.pdf
[5] R. K. Chellappa, "Intermediaries in cloud-computing: A new computing paradigm," in *Proc. INFORMS Annu. Meet.*, Oct. 1997.
[6] N. A. Sultan, "Reaching for the 'cloud': How SMEs can manage," *Int. J. Inf. Manage.*, vol. 31, no. 3, pp. 272–278, 2011.
[7] F. Durao, J. Fernando, S. Carvalho, A. Fonseka, and V. C. Garcia, "A systematic review on cloud computing," *J. Supercomput.*, vol. 68, no. 3, pp. 1321–1346, 2014.
[8] C. Chapman, W. Emmerich, F. G. Márquez, S. Clayman, and A. Galis, "Software architecture definition for on-demand cloud provisioning," *Cluster Comput.*, vol. 15, no. 2, pp. 79–100, 2011.
[9] W. Christof *et al.*, "Cloud computing—A classification, business models, and research directions," *J. Bus. Inf. Syst. Eng.*, vol. 1, no. 5, pp. 391–399, 2009.
[10] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. IEEE 39th Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2015, pp. 53–59.
[11] C. Vallati. *Making Fog Computing Real-Research Challenges in Integrating Localized Computing Nodes into the Cloud*. Accessed: Jun. 12, 2017. [Online]. Available: https://blog.zhaw.ch/icclab/making-fog-computing-real-research-challenges-in-integrating-localized-computing-nodes-into-the-cloud/
[12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[13] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 2, no. 1, pp. 19–26, 1998.
[14] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, 2010, pp. 301–314.
[15] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.
[16] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.
[17] Y. Wang, "Definition and categorization of dew computing," *Open J. Cloud Comput. (OJCC)*, vol. 3, no. 1, pp. 1–7, 2016.
[18] K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," *Open J. Cloud Comput.*, vol. 2, no. 1, pp. 16–24, 2015.
[19] S. Ristov, K. Cvetkov, and M. Gusev, "Implementation of a horizontal scalable balancer for dew computing services," *Scalable Comput., Pract. Exper.*, vol. 17, no. 2, pp. 79–90, 2016.
[20] Y. Wang, "Cloud-dew architecture," *Int. J. Cloud Comput.*, vol. 4, no. 3, pp. 199–210, 2015.
[21] T. Ozsu and Valduriez, *Principles of Distributed Database Systems*. New York, NY, USA: Springer, 2011.
[22] U. F. Minhas, S. Rajagopalan, B. Cully, A. Aboulnaga, K. Salem, and A. Warfield, "RemusDB: Transparent high availability for database systems," *VLDB Endowment*, vol. 4, no. 11, pp. 29–45, 2011.
[23] Y. Wang and Y. Pan, "Cloud-dew architecture: Realizing the potential of distributed database systems in unreliable networks," in *Proc. Int. Conf. Parallel Distrib. Process. Techn. Appl. (PDPTA)*, 2015, pp. 85–89.
[24] B. B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *Proc. 19th Int. Conf. Data Eng. (ICDE)*, Mar. 2003, pp. 49–60.
[25] B. Yang and H. Garcia-Molina, "Comparing hybrid peer-to-peer systems," in *Proc. 27th Int. Conf. Very Large Databases (VLDB)*, 2001, pp. 561–570.
[26] O. Ulusoy, "Research Issues in peer-to-peer data management," in *Proc. 22nd Int. Symp. Comput. Inf. Sci.*, Nov. 2007, pp. 1–8.
[27] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu, "Data management for peer-to-peer computing: A vision," in *Proc. 5th Int. Workshop World Wide Web Databases*, 2002, pp. 89–94.
[28] S. Sehlhorst. (Nov. 2008). *The Economics of Software as a Service (SAAS) vs. Software as a Product. Pragmatic Marketing*. Accessed: Jun. 29, 2017. [Online]. Available: http://pragmaticmarketing.com/resources/the-economics-of-software-as-a-service-saas-vs-software-as-a-product
[29] R. Rajala, M. Rossi, and V. K. Tuunainen, "A framework for analyzing software business models," in *Proc. 11th Eur. Conf. Inf. Syst. (ECIS)*, Naples, Italy, Jun. 2003, pp. 1614–1627.
[30] K. Popp, "Software industry business models," *IEEE Softw.*, vol. 28, no. 4, pp. 26–30, Jul. 2011.
[31] S. A. Mokhtar, S. H. S. Ali, A. Al-Sharafi, and A. Aborujilah, "Cloud computing in academic institutions," in *Proc. 7th Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, 2013, pp. 2:1–2:7.

PARTHA PRATIM RAY received the B.Tech. degree in computer science and engineering and the M.Tech. degree in electronics and communication engineering, with specialization in embedded systems, from the West Bengal University of Technology, Kolkata, India, in 2008 and 2011, respectively. He is currently a full-time Assistant Professor with the Department of Computer Applications, Sikkim University, Gangtok, India. His research interests include Internet of Things, pervasive bio-medical informatics, and wearable computing. He was elected as a member of the IEEE CSI Executive Council of India in 2014 and 2015. He received the VIRA Young Scientist Award and Bharat Vikas Award in 2017, for outstanding contribution in his field.