デジタル庁

「国・地方ネットワークの将来像及び実現シナリオに関する検討会」

第二回会議 (2023/11/02)

提出資料 (意見書)

2023/10/30

登 大遊

目次

第１章　[redacted]提出案の方針に関する意見

第１節　要点

１　要点

[redacted]の[redacted]で示されている方針について、下記記載のリスクとそのようなリスクに対する緩和策を確実に実施することを前提条件として、基本的に、賛成である。

２　人的・組織的な面におけるリスクと緩和策

[redacted]には、人的・組織的な面における重要なリスクがいくつかあるが、いずれも、いくつかの緩和策により、緩和が可能である。下記のとおり、具体的な緩和策を提案する。

(ア)　「厳格なシステム」([redacted]で示されている方針のような計画的なシステムのこと。以下同じ) は、一般的コンピュータ・リテラシを有する 95% のユーザーを収容するためのものであり、セキュリティ事故を防ぐため、自由度に対する一定の制約が正当化され、許容され得る。しかし、すべての面がこの方針で統制されるとなると、極めて能力の高い 5% のユーザーの人材育成、試行錯誤、セキュリティ能力向上、新技術発育を、現在よりも著しく阻害するおそれがある。そこで、このリスクを緩和するため、これらの高水準人材のために必須となる、自由なシステム (第 2 章) を用意するべきである。

(イ)　95% のユーザーに対する各種統制 (端末、アプリ、通信に対する統制) を施す場合であっても、過度な制約による人材育成の制約、セキュリティ・リテラシの低下、各自の頭脳を用いた創造的な問題解決方法の発案の阻害等が生じないようにするため、以下の対策を行なうべきである。

① 常に、制約が必要な最小限度に保たれているかどうかを、システム管理

者の管理行為を監督する外部的監視の仕組みを用意すること。

② 当該監視の仕組みは、少なくとも当該委員会の委員の水準またはそれ以上の水準の自由かつ秩序のある思想を有する合議監督的仕組みによって構築されること。

③ 95% のユーザーの個々の多様な技術レベルや事情に合わせて課せられる制約の質や強弱は、適格・迅速かつ柔軟に設定される運用が保障されること。

④ 制約を受けるユーザーからの事情の申し出および不服の申出には、システム管理者的視点・短期的視点のみでなく経営的視点・長期的視点・技術開発および人材育成的視点を総合した考慮のもと適切に対応する権力体制が維持されること。

(ウ) デジタル庁による中央集権的管理がすすむと、強い権限を有するシステム管理権者 (技術者権力) が、デジタル庁に出現し得る。そのような者は、統治が IT システムに依存しつつある日本国全体の中で、省庁横断的に、強い権力を振るうことができてしまう。そのような権力者に対する衆人環視と文民的統制を徹底し、強大な技術者権力の出現の抑制を確実に行なう必要がある。これにより、日本国の統治の長期安定が実現できる。

(エ) デジタル庁が、決して、互いに独立した各省庁・各地方自治体のクラウド領域に対する特権的管理権を有する状態とならないことが必要である。これにより、各省庁・自治体の多様な考え方を尊重することができ、独立した思考を妨げることを予防でき、各組織の優秀な人材の思考を最大限に活用した行政運営が可能となる。

(オ) ファイアウォールや HTTPS 中間者攻撃型プロキシシステムにおけるインターネット通信制御におけるポリシールールは、必ず、組織ごとに個別に決定・管理されるようにし、各省庁・各自治体の組織ごと、職員ごとの価値観や思想の多様性を制約しないようにすることが必要である。

(カ) デジタル庁等の国の一部のシステム管理権力が、国全体の省庁・自治体の端末ログ、Web アクセスログを集約・検索可能とするシステムは構築しな

いことが重要である。これにより、地方自治体や各省庁などの独立組織におけるインターネットやコンピュータの利用に関する事柄についての、各組織における自律性が尊重される。

3　技術的な面におけるリスクと緩和策

提案には、技術的な面 (特にセキュリティ面) における重要なリスクがいくつかあるが、いずれも、いくつかの緩和策により、緩和が可能である。下記のとおり、具体的な緩和策を提案する。

(ア) パブリッククラウド環境において、数百万社の企業が同じクラウド基盤システムを利用していれば、脆弱性を発見され攻撃される危険がはね上がるので、クラウド利用にあたっては、そのリスクに係る予見義務と結果回避義務を果たした形のシステムを構築することが必要である。これにより、パブリッククラウド側の基盤システムの脆弱性が原因で大規模なセキュリティ侵害が発生しても、日本政府および地方自治体は、その被害による影響を避けることができる。

(イ) パブリッククラウドの事業者内秘匿基盤ソフトウェアの脆弱性は、ユーザーがバイナリすら見ることができず、衆人環視が困難であることから、当該基盤上の部分はゼロトラスト原則上の Untrusted Zone として扱い、その上に暗号化等の論理的手法を加えることにより、安全な極小化された Trusted Zone を実装することが必要である。これにより、ゼロトラスト原則とパブリッククラウドの利用とが両立できる。

(ウ) ゼロトラスト原則の本質を定義に戻って理解し、まずは従来のオンプレミスシステムにゼロトラスト原則に適合する設定を加えることによりこれを実現することを試み、加えて、ゼロトラスト原則と反対の結果を生じさせる表見的ゼロトラスト製品は敬遠することが必要である。これにより、表見的ゼロトラスト製品を用いてしまい、逆にセキュリティが低下してしまうという、本末転倒な現象を予防できる。

(エ) パブリッククラウドの機密性・完全性・可用性の突然の喪失への各種対処方法を予め講じる必要がある。全データは必ず頻繁にローカルにバックア

ップをとり、クラウドサービス停止時においては、手元の環境 (サーバーやネットワーク) を用いて、短時間でオンプレミスにおいて同等システムを再開することができるような準備を、常に怠らないことが必要である。これにより、パブリッククラウド基盤のセキュリティ侵害時や停止時においても、慌てることなく、先ほどまで動作していたシステムをほとんどダウンタイムなしで継続稼働させることが可能となる。

(オ) パブリッククラウド上のストレージへのデータのアップロード時には、呼び出し元アプリケーション側の CPU において、クライアント側暗号化を確実に施すこと (仮にパブリッククラウド基盤が攻撃者によって掌握されていた場合でも、攻撃者が平文にアクセスできない措置の具備) が、常に必要である。これにより、パブリッククラウドのストレージにかなり高度な機密情報 (個人情報等) をはじめて安全に保管することができるようになる。

(カ) 端末・OS の統合管理のシステムは本質的危険 (わずかな誤りによる日本全体の行政職員の端末停止のリスク) を有するので、一極集中・単一システム依存を避け、組織間で安全に分散させ、システムにおいても、多様性を実現することが必要である。これにより、サイバー攻撃者による端末・OS の統合管理システムの掌握時にすべての行政端末が一斉にロック・データ消去されるおそれがなくなり、世界的な深刻なサイバー攻撃発生時においても、日本の行政機関は、被害を免れることができ、日本国の統治を継続できるようになる。

(キ) 電子メールサーバーシステムは、従来のオンプレミス同様の多様性による強靱的セキュリティを、クラウド環境でも IaaS を用いて実現し、従来よりもセキュリティレベルを低下させないことが必要である。これにより、インターネットと接続された電子メールシステムであっても、サイバー攻撃に対して安全にメールサーバーを運用でき、高度なクラウドサービスを狙ったサイバー攻撃からメールシステムを保護することができる。

(ク) 重要な業務システムで用いる ID 認証基盤の構築のためにクラウドサー

ビスを利用する場合は、ソフトウェアそのものに対する衆人環視が確保されているソフトウェアを IaaS で用いること (SaaS の ID 認証基盤の利用は、可能な限り避けること) が必要である。これにより、ID 認証基盤を狙ったサイバー攻撃を回避することができ、すべての ID 認証利用システムが保護している大量の資源に攻撃者がアクセスしてしまうことを根本的に予防することができる。

(ケ) 重要なクラウド上のデータベースは SaaS、PaaS ではなく IaaS で構築することが必要である。これにより、データの破損のリスクや機密性の喪失のリスクを大幅に引き下げることができ、不具合が発生した場合でもユーザー組織自ら原因を特定して回復することが可能となり、システム障害が発生するリスクを予防できる。

(コ) HTTPS 中間者攻撃型プロキシサービスをやむを得ず構築する場合は、オンプレミス型または IaaS 型を利用することが必要である。これにより、ゼロトラスト原則における Trusted Zone を極小化でき、認証セッション文字列 (Cookie)、電子メール、ワンタイムパスワード認証のための基礎となる秘密鍵の QR 画像、システム管理者特権等をサイバー攻撃者に奪取されるリスクを最小限にすることができる。

(サ) ネットワーク機器、SDN 装置、ファイアウォール等は、多様性が重要であり、世代もファームウェアのバージョンも異なることが望ましく、特に、全自治体に適用するインターネットベース・クラウドベースのネットワーク機器の中央管理システムは避ける必要がある。このような多様性実現によるセキュリティ対策は、セキュリティを高めることはもちろん、各省庁・自治体の職員の方々のセキュリティ能力と IT 能力を組織的・飛躍的に高めることにつながる。

(シ) クラウドを用いた AI サービスの利用については、AI サービス提供事業者からの偏向攻撃に対する安全措置を講じることが必要である。これにより、AI を政府業務で日常的に比較的安全に利用できるようになる。

これらのリスクと緩和策に関しては、以下でより詳しく述べる。

第 2 節　人的な面におけるリスクと緩和策

1　厳格なシステムにおけるユーザーに対する制約の正当化とリスク緩和策の具備

この検討会で提出されている、GSS の事例、LGWAN との相互接続、ガバメントクラウド、セキュリティ製品、ゼロトラスト製品の導入といった事柄は、第 1 回会議において議論をさせていただいた (1) 「厳格なシステム」と (2) 「自由なシステム」の発想においては、(1) 「厳格なシステム」に近い。こういった「厳格なシステム」は、限られた時間で、一応のそれなりに安全性をもって、一応確実に実現させる必要がある。ただ、(1) で実現できる、既存製品ベースのシステムは、本質的には実はセキュアではない点も多い。現在の製品市場は、真実の安全性を実現するに足るレベルには、未だ達していないからである。そういう水準に達する製品群は、これから 10 年、20 年かけて、おそらく日本人たちが生み出して (歴史をみると、米国で開花した技術は、30 年後に日本で発展する具合であるから)、うまくいけば、2030 ～ 2040 年頃に、いよいよ、世界中で普及されてゆくであろう (これについては、第 2 章で述べる)。しかし、本 IT 計画は、それよりも前の、わずか数年後の 2025 年頃、2030 年頃を目標とするものである。理想的な完成度を有するセキュリティ製品等が未だ存在しない状態であっても、とにかく計画を進めなければならない。このやむを得ない状況下であるという事実には、誰もが同意するであろう。このような状況では、お金で問題を一応解決するということは、合理的である。時間的制約から、可能な限り、第三者である SIer や既製品製造業者に、責任をとっていただく形で進めることも、やむを得ない場合がある。今はまさにそのような場合であると思われる。ただし、第三者の未完成な製品を導入した結果、欠陥があり、セキュリティ事故が生じたとして、それにより損害を被った被害者 (たとえば、国民) から賠償請求があっても、国や自治体は、製品提供者 (第三者) に過失があったのだといって、自らの国家賠償責任を免れることはできない。もちろん、国や自治体は、製品提供者に対して求償をすることはできる。しかし、たいていの場合、パブリッククラウド事業者や製品提供者においては、使用許諾契約書等が注意深く作成してあって、わずかな限度額以上の賠償はしない等と記載さ

れている。結局国や自治体は賠償を受けられないことも多い。だから、実際には、問題は解決されていない。この類型の問題については、後に詳しく述べる。しかしながら、それは随分先の将来に突然に発生する問題であり、その時には責任の所在はもはや多数の集団に吸収されていて、個人に集中することはない。そしてまた、重過失や故意がない限り、行政職員個人が求償されることはない。個人的視点では、これは安全である。国民の視点から見れば、問題は結局解決されていないのだが、IT の問題に限らず、国の問題解決方法というものは、たいていこのようなものである。これは、時間的制約から仕方がないことであり、仮に現実時間内に真に問題を解決する改善方法を述べよという具合になっても、応じることができる材料は誰も持ち合せていない。したがって、このような問題を指摘することは一見無意味である。しかし、十分に技術が進歩したかなり将来の時点で、いずれ誰かがこの議事録を発掘して、昔このようなことが一応指摘されていたという事実を知って、過去の行政 IT の雰囲気はこのようななかなか面白いものであったのかという歴史的考察に、おおいに役立てるかも知れない。だからここにこのように書いて、この会議に提出しておこうと思う。

さて、厳格なシステムは、一般的ユーザーとしてのコンピュータシステムの利用職員が、製品やすでに現実化されている技術を組み合わせて、何らかのシステムを一応作ることができる程度の、中級程度のエンジニアにとって、それなりに有益である。これらの初級・中級程度のユーザーやエンジニアは、一定の頻度で、深刻な間違いを引き起こし、判断能力も、責任を取る能力も発達途上にある。こういったユーザーの方々が、現状は、行政機関の 95% 以上を占めている。現在のクラウド製品、ネットワークセキュリティ装置、ゼロトラストシステム等の既製品は、ユーザーの自由に制限を課す機能が豊富にある。これらの製品は、前述のとおり、かなり発展途上で粗が目立ち、高度なユーザーから見ればセキュリティホールが豊富に存在するので、実のところあまり無意味である。しかし、95% のユーザーにとっては十分に強力な自由に対する制約となり得る。そこで、その制約を正当化し得るかどうかを検討してみよう。確かに、そういった制約がない完全な自由環境下 (たとえば、自宅コンピュータ・ネットワークや先進的 IT ベンチャー企業の開発者向

けコンピュータ・ネットワーク等) と比較して、大きな制約がある場合は、技術的な試行錯誤が一定程度困難になり、人材育成の速度も劣化し得る。そうすると、あまり規制を加えないほうが良いというようにも見える。しかし、それは早計である。中途半端な中級ユーザーが自らの技術力を過信してリスクの高い行動に出た場合に生じる問題の影響は思いのほか大きく、日本的無謬性が要求されるわれら行政機構においては、その種の問題 (たとえば、結果回避可能性が十分にあったような不注意によるセキュリティ・インシデント) の発生は、しばしば、致命的な結果となり得る。そのようなインシデントを理由として、IT に関する試行錯誤を一切禁止すべきという風な声が上がり、これが支配的となると (日本では、そのようになりやすい)、試行錯誤が逆に抑制され、95% に対する人材育成が困難となり、回復不能な損害が発生するおそれが高い。よって、そのような損害を予防するため、こういった 95% の発展途上のユーザーに対して、既製品を活用して「厳格なシステム」を構築し、必要な限度で制約を課すという行為には、合理性があり、正当化し得ると考えられる。ただし、95% のユーザーに対する制約であっても、これらのユーザーが自らの水準に合わせた創意工夫によって IT 能力を高め、試行錯誤を行ない、自らと組織全体の人材育成を実現しようとする合理的で健全な努力を不当に阻害する程度に強度の制約を課してはならないことは、もちろんである。システム管理者というものは、ややもすると、そういった IT 人材育成的視点よりも、自らの管理を安楽にし仕事量を削減することを優先しがちである。そして、大抵そのような強権的制約を好むシステム管理者の IT 能力は、95% の集団の中でも比較的中程度なのである。このようなシステム管理者たちの自らの仕事量の削減への優先的努力加重は、組織に対する、また、日本国に対する長期的視点における利益相反である。自らの安楽の利益との引換えに、全員の人材育成的価値・技術力向上の価値を犠牲にしているためである。人間の心は、弱いものなので、どうしても管理権を有するとそのような楽な方法、圧政を敷く方向に傾いてしまうのである。各個人の力では、これは予防が困難である。そこで、これは、組織的・仕組み的に、予防しなければならない。よって、「厳格なシステム」において様々な制限システムを活用して 95% のユーザーに対してシステム管理者が画一的制約を施すことを許容す

る場合、① 常に、制約が必要な最小限度に保たれているかどうかを、システム管理者の管理行為を監督する外部的監視の仕組みを用意すること、② 当該監視の仕組みは、少なくとも当該委員会の委員の水準またはそれ以上の水準の自由かつ秩序のある思想を有する合議監督的仕組みによって構築されること、③ 95% のユーザーの個々の多様な技術レベルや事情に合わせて課せられる制約の質や強弱は、適格・迅速かつ柔軟に設定される運用が保障されること、④ 制約を受けるユーザーからの事情の申し出および不服の申出には、システム管理者的視点・短期的視点のみでなく経営的視点・長期的視点・技術開発および人材育成的視点を総合した考慮のもと適切に対応する権力体制が維持されること、の 4 点を条件とするべきである。これらの安全措置が施されている限り、現代の未完成な水準の市場のセキュリティ製品、ネットワーク製品を用いてユーザー環境に一定の制約を施しても、短期的・中期的には、それなりに安全である。2030 年頃まではそれで良いと考えられる。

2　強大な技術者権力の出現の抑制 ― いよいよ権力集中するシステム管理権者の権限濫用を予防するための衆人環視と文民的統制の徹底の必要性

国および地方自治体のさまざまなシステム部分を統合的に調達、構築、運用する場合、ネットワーク、セキュリティ、クラウドシステムの管理権者 (技術的人材が担うであろう) の権力は最小限にし、その権限行使は、その技術者よりも高い経営能力、法的能力、技術能力を分散して有する官僚たちと有識者たちによって、常に、衆人環視しなければならない。これは、最上位の特権的管理者である技術者に対しても、例外なく、適用されなければならない。日常は、彼ら技術者たちを一応信用してよい。しかし、彼ら技術的特権者は、目を離すと、かならずその技術的権力を拡大しようとする。そのうち、過失または故意で事故を引き起こす。従来は、これは組織ごとに発生したが、デジタル庁がさまざまな組織の部分を情シス的に担うと、その権力は最大化し、歴史上みられなかった程度の、技術者たちの権力体制 (技術者天国、技術者たちがそれ以外の統治領域に影響を及ぼす危険) が確立されてしまうおそれが生じる。これが蓄積され、最大級のシステム管理権力濫用リスクが生

じる。あるときにそれが現実化し、行政 IT の上に依存している各種のものごとが甚大な影響を受ける。昔は、軍部の暴走が最大のリスクであった。今は、技術者の暴走が、最大のリスクである。この 2 つの危険性は、本質的に同じである。技術者たちは、放っておくと、広い視野で当然に検討しなければならないさまざまなリスクを過小評価し、情報をゆがめて人文系経営者たちに伝えてしまう傾向がある。また、社会発展においてはコンピュータ技術以外にも幅広い領域における同時的価値向上が必要だが、権力傾向のある技術者たちは、自らの周辺の技術領域が他の領域よりも随分優先して投資されるべきであるという、立場に基づいた利益誘導を、それとなかなか気付かれることなく、人文系官僚、政治家達を操って引き起こすことができてしまう。人文系官僚、政治家たちは、こういった技術者たちになかなか対抗することができない。このような技術者権力集団がひとたび生じてしまうと、国全体が、大変な危機にさらされる。これを防ぐ方法をあらかじめ具備しなければならない。監視機構は通常は形骸化に近い形であってもよいが、いざというときに、技術者権力者集団に対して統制ができる権限を複数人で分散して有していなければならない。イメージでいうと、警察権力に対する公安委員会のようなもの、軍隊に対する文民統制のようなものである。システムに関する権限を掌握した現代型技術者たちは、物理的強制力をはたらかせることができる警察権力に相当するから、とても危険である。特権を有する技術者たちは、自らの独自の利益を、共同の利益・公的利益より優先させてさまざまなことを決め、水面下で行動する傾向がある。経営的地位にいる管理職たち、政治家たちは、技術者権力に、今から、十分対応できるようにしなければならない。今は、日本の行政系人文系管理者たちの一定の割合は、技術者たちの集団を、純粋で純技術的なものであると無意識に信じてしまっているところがあるが、技術者たちは、必ずしも、そのような純粋なものではない。技術者の欲する権力に対する意欲と、人文系の有する権力に対する意欲とは、同等程度である。人文系の権力集中を抑える手法は長年研究され適用されてきたが、高度な技術者に対しては、コンピュータやネットワークが登場して社会がその上で動くようになってからまだ間もないこともあり、彼らを抑制して程良い水準に制御する方法が、未発達である。これらを研究して発達させ、技術者に対する無防備性を

有する社会全体を安全な方向に引っぱっていくことが、社会の存続・発展のために、極めて重要である。そのような体系化された安全の仕組みの出現には、まだ時間がかかるし、われわれのような人文社会系・技術系の公的集合体の存在意義は、それを発展させてゆく立場という重要な責務という点に見出されるのである。

3　デジタル庁が各省庁・各地方自治体のクラウド領域に対する管理権を有してはならない ― 地方自治の本旨と多様性によるセキュリティが失われることを予防する

デジタル庁 (国) が各地方自治体にクラウドサービスを勧めるのは良いことである場合もあるし、各地方公共団体のシステム管理者が困っている場合は、使い方をサポートするのは良いことである。しかし、デジタル庁が各地方自治体のシステムに対する管理権を有してはならない。すなわち、デジタル庁は、いかなる場合であっても、各地方自治体が利用するクラウドサービスにおける、ユーザー組織側の特権管理者としての地位を有してはならない。デジタル庁は、各地方自治体が個別的かつ一時的に明示的に与えた場合を除き、地方自治体の利用するクラウドサービスの管理者アカウントを用いて地方自治体が使用・管理するクラウド領域にログインしてはならない。そのようなログイン可能なアカウントを有してもならない。

なぜならば、国と地方自治体とは異なる法人であり、各地方自治体が行政活動に利用する IT システムの管理権は、各地方自治体固有のものであるためである。各地方自治体のクラウド領域は、各地方自治体がアクセス管理者である。各地方自治体がアクセス管理者である以上、国は、各地方自治体の許可なく、各地方自治体のクラウド領域に管理権限またはユーザー権限でログインしたり、その他の抜け道を使って管理者権限またはユーザー権限を用いた振る舞いを行なってはならない。これは通常は不正アクセス禁止法に違反する。

ところが、仮にデジタル庁とクラウドサービス事業者とが、各地方自治体に代わって、ボリューム・ディスカウントを狙って一括契約するとしたならば、その契約の書き方、各地方自治体との間で締結する利用規約等によっては、上記のような、本来不正アクセス禁止法等で保護されている各地方自治体固有のクラウド領域について、デジタル庁の職員が、各地方自治体の明示的かつ個別の許可なく管理権限

またはユーザー権限でアクセスしても、不正アクセス禁止法に問われない脱法領域を作ることができてしまう。これが、最も危険なことである。各地方自治体の内部事務関係は、原則として、各地方自治体が独立して自ら決定・処理できる。これは憲法 92 条で地方自治の本旨として保障されている。団体自治の原則は、地方自治体のみが、他の干渉を受けずに、地方自治体の業務を実施することができるという原則である。各地方自治体のコンピュータ・システムは、その地方自治体の権限のある職員のみがアクセスできなければならない。明示的・個別の許可を得た場合以外、それ以外の個人や法人がアクセスできてはならない。他の地方自治体がアクセスしてはならない。もちろん、国もアクセスしてはならない。国と地方自治体とは、異なる法人である。デジタル庁職員が管理権を入手して、各地方自治体のシステムに、アクセスしてはならない。各地方自治体がどのような文書を作成し、保管し、どのような計画を立てようとしているか、決して見てはならない。仮にデジタル庁とクラウドサービス事業者とが、各地方自治体に代わって、ボリューム・ディスカウントを狙って一括契約するとしても、クラウド領域の管理については、そのようなのぞき見が絶対に不可能な契約と運用体制としなければならない。クラウドサービス事業者によっては、1 つの契約で複数のテナントを分割運用することができる場合がある。しかし、契約代表者がすべてのテナントにアクセスできてしまうような構造になっている場合がある。これは、国と地方のクラウドの共同調達においては、許されない。国が契約代表者として特権アカウントを有してしまうと、各地方自治体のテナントの中身を、個別具体的承諾なくして、包括的に、いつでも、予告なく、自由に見て回ることができてしまう。その可能性があるというだけで(すなわち、各地方自治体の職員が、自分たちのクラウドに置くデータはデジタル庁の管理権限を有する職員がいつでも見ている可能性があると認識するだけで)、全地方自治体のクラウド活用は阻害される。国に見られても良いというデータだけがクラウドに置かれるようになり、それ以外のほとんどのデータはクラウドには置かれない結果となってしまう。その結果、クラウドが活用されなくなってしまう。また、その状態で、仮に国がクラウドの利用をさらに押し付けると、各地方自治体の職員は、今度は、国に迎合した業務を行なうようになる。すなわち、地方自治体

が、国の方針に何でも協力的となり、何でも国に従ってしまうような心理状態になってしまう。これは、地方自治体の存在による国政の多様性を損傷する。地方自治の本旨を侵害してしまい、回復不能なレベルに陥るおそれがある。地方自治体の存在意義は、ある面では国の方針に協力し、別の面では国の方針に異議を唱え、場合によっては従わないという、独立・自律の点にある。憲法で地方自治が制度上保障されているのは、主権者たちが、国の存立のためには地方自治の本旨の保障が極めて重要であると考えたためである。われわれは主権者たちの定めた各地方自治体と国との独立・対等の原則を遵守しなければならない。

デジタル庁が、決して各省庁・各地方自治体のクラウド領域の管理権を決して保有しない (管理権をあえて利用しないのではなく、そもそも、管理権を有していない) 状態は、安全なサイバーセキュリティの実現にも必須である。すなわち、日本の統治の安全性・強靭性は、国と地方自治体とが異なる管理権を有していることによって実現されているが、仮に国 (デジタル庁) がすべての地方自治体のクラウド領域に対する特権的アクセスが可能なアカウントがあれば、それをサイバー攻撃者が奪取した場合の被害は、国の存立にとって致命的な結果となるためである。そのような超強力な管理者権限の存在は、ゼロトラスト原則にも反する。ゼロトラスト原則では、暗黙的 Trusted Zone を可能な限り極小化しなければならない。そのような超強力な管理者権限は、その認証クレデンシャルが暗黙的 Trusted Zone そのものであるといってよい。そのような極めて強力な管理者権限が、全省庁・全自治体の全クラウド領域に対するものであれば、その認証クレデンシャルを有する者は、日本の統治機構のほとんど全てのクラウドシステムに対する Trusted Zone にアクセスできることになってしまう。サイバー攻撃者が外部にいる場合で、そのクレデンシャルを奪取した場合も、また、国の内部の技術者が権力欲に駆られてそのクレデンシャルを悪用した場合も、致命的な結果が発生する。統治において最も危険な現象、何としてでも避けなければならない危難は、特権のある技術者の暴走である。このような、国の統治機構が 1 つの誤りによって倒壊してしまう危険を予防するために、デジタル庁が、決して各省庁・各地方自治体のクラウド領域の管理権を絶対に有さない (管理権をあえて利用しないのではなく、そもそも、管理権

を有していない) 状態を維持する必要がある。従来より、省庁システムや地方自治体システムは、分離・独立して構築されてきたので、このような多様性・免疫性を自然に備えていた。ゼロトラスト原則は、このような多様性・免疫性をさらに拡大し強化するものであり、たいへんに望ましいものである。ところが、クラウド化による画一化・集中化は、このような多様性・免疫性をむしろ弱体化させ、危険性が増すこともある、ハイリスク手法である。ここにおいて、デジタル庁が各省庁・各地方自治体のクラウド領域に対する管理権を有することは、その危険性を限りなく増大させ、日本建国以来最大の危険が生じるおそれがある。従来の危険は、国の外側の要因で生じてきた。通常のサイバー攻撃も、たいていは外部要因で生じる。ところが、技術的管理権限の集中は、内部要因による人的暴走故障が原因であり、致命的な被害が発生してしまう。これを予防する唯一の方法は、そのような全組織にわたる特別な権限を決して作成しないことである。いかに信用できるように見える技術者にも、そのような特別な権限を与えてはならない。そのような特別な権限を技術上作成できるとしても、その作成行為そのものを、厳重に禁止しなければならない。これを予防するための最良の方法は、クラウドサービス事業者と仮に一括契約するとしても、クラウドサービス事業者の側で、絶対に、横断的特権アカウントを発行できない仕組みで、完全なテナント間分離を実現してもらうことである。これは極めて容易に可能である。これにより、特権アカウントの奪取による致命的被害は避けることができる。もっとも、クラウドサービス事業者そのものの内部の特権アカウントのサイバー攻撃者による奪取による致命的被害は避けることができないが、これはクラウドサービスを利用する上でやむを得ないものである。これは本文書の後の部分で述べているようなさまざまな方法で緩和するしかないことに留意する必要がある。

4　ファイアウォールや HTTPS 中間者攻撃型プロキシシステムにおけるインターネット通信制御は、国全体で共通化・一元管理されてはならない。ポリシーは、必ず、組織ごとに個別に決定・管理されなければならない

ファイアウォールや HTTPS 中間者攻撃型プロキシシステムにおけるインター

ネット通信制御が必要となる場合がある。このうち、HTTPS 中間者攻撃型プロキシシステムは、セキュリティ上の問題がある (後述する) ため、できる限り、避けるべきである。しかし、特定の目的で、必要最小限に、やむを得ずこれを導入する場合においては、それぞれの組織の HTTPS 中間者攻撃型プロキシシステムはできるだけ多様なものを導入する必要がある。加えて、通信ポリシーのルールは、組織間で共通のルールに依ってはならない。特に、中央権力である国 (デジタル庁) が、地方自治体の通信ルールを勝手に規定してはならない。国 (デジタル庁) が、地方自治体の通信ルールを勝手に規定し、これを中央からアップデートし変更することができるとなると、次のような重大な脅威が発生する。すなわち、地方自治体の職員が、いかなる Web サイトを見て、いかなる情報収集を行ない、いかなるアイデアを構築するかは、原則として、各地方自治体の長とその責任のある職員たちに委ねられる。そのルールは、地方自治体ごとに多様なものでなければならない。仮に国が中央からインターネット検閲ポリシーを決定し、これを地方自治体に対して配信したならば、国の思想が支配的となる。たとえ強制されなくとも、地方自治体の側は、国に忖度をして、国が配信する検閲ルールをすすんで導入してしまうおそれがある。地方自治体が自らの頭脳でルールを考え、独自のルールを制定しようとする契機を奪ってしまうことになる。これにより、国の方針に従う従順な地方自治体とその職員たちが育成されてしまうおそれがある。これは、極めて危険である。地方分散された民主主義体制が危機に瀕する。憲法で規定されている団体自治の制度的保障に反する結果が生じる。このような重大な統治上の問題を予防するためには、国が各地方自治体に対してインターネット検閲システムのポリシーに関する中央集権的関与を行なってはならないことが重要である。

そのことは、同様に、国の中の省庁間においても当てはまる。たとえば、デジタル庁は国の府省の下部組織であるが、システム管理者権限を濫用して、他の省庁すべてのインターネット検閲ポリシーを操作することができたならば、省庁間の思想の多様性に影響を与えることができてしまい、強靭な意思決定基盤である優秀な各省庁の官僚のさまざまな思想、アイデアを単一化できてしまう危険が発生する。インターネット閲覧の安全という大義名分の名の下に、多数の職員の思想を一定の方

向に誘導しようとする活動が可能になってしまう。これは極めて危険である。省庁間、自治体間の、決して一致することがない多様な考え方の維持が、安定した発展的な国政の運営のために、極めて重要である。各府庁のインターネット検閲ポリシーは、そもそもそのような規制が必要ない程度に職員のセキュリティ・リテラシが高まることが理想的であるが、それまでの間にやむを得ず必要な場合であっても、各府庁の大臣・長官の責任のもと、各府庁の担当者が決定するものでなければならない。本来、各府庁の大臣・長官の責任で決定されるべきネットワークセキュリティポリシーを、デジタル庁のような一役所が、権威のあるものとして策定し、これを他省庁に対して押し付けてはならない。各省庁は、独立・対等であり、自律性が何よりも重要である。ネットワークセキュリティポリシーの中央集権的配布は、その自律性を喪失させるおそれがある。

5　システム管理権力が、ログを集約し検索可能としてはならない

システム管理権力が、日本国の各省庁や地方自治体のすべてのユーザーの端末操作ログや Web サイトへのアクセスログを集約し検索可能とすることは、現に禁止しなければならない。いかなる組織のいかなる職員が、いかなる時間帯に、どの Web サイトを閲覧したかというログが、国の特定の組織 (例えば、デジタル庁) に集約されるとしたら、それは、そのログを検索・閲覧できる権限を要する管理的職員やその組織に絶大な権力が生じてしまうことを意味する。その権力が濫用されると、政府内において特定の個人または集団 (特に、技術者集団) に大きな力が生じてしまい、行政部門において、公正な意思決定ができなくなる。加えて、そのような集約・検索が、国全体で一元化されてなされているというだけで、本来独立・対等な多数の省庁および地方自治体の各職員の組織的な独立したアイデアの確立や思索を妨げてしまう。

端末や Web アクセスのログを集約して記録することは、従来より行なわれてきたことであるが、これは、各省庁、各地方自治体でそれぞれ独立に行なわれてきたのである。各省庁の長は大臣・長官であり、各自治体の長は首長である。これらの各責任者による分散的管理の下であれば、端末操作や Web アクセスの記録は一応安全である。これらの責任者の枠を超えて、デジタル庁等の国の特定部局が、指揮

命令体系を超えて、すべての官公庁や自治体の端末ログ、Web アクセスログを一括で保管し検索できる体制は、決して構築してはならない。

第 3 節　技術的な面におけるリスクと緩和策

1　概論

(1)　入門編 ― 意思決定者 X と技術者 Y のゼロトラストとパブリッククラウドに関する会話

意思決定者 X「本システムは、重要なので、ゼロトラストの原則を適用したい。そこで、今日は、Trusted Zone の極小化とセキュリティ担保を議論しよう。現在の責任体制は、どのようになっているのか。」

技術者 Y「パブリッククラウド事業者 A が提供する基盤上のサービス B を利用する予定である。B には Trusted Zone が存在し、ユーザーとクラウド事業者 A との責任分界モデルにより、この Trusted Zone のセキュリティの責任はクラウド事業者 A が負うことになっている。」

意思決定者 X「その Trusted Zone が攻撃者に破られたら、われわれはひとたまりもない。クラウド事業者 A は、その Trusted Zone の安全性をどのように担保しているのか。」

技術者 Y「その体制は、クラウド事業者 A の企業秘密であり、われわれはそのソースコードもバイナリも構築手順書も確認できない。しかし、クラウド事業者

A の技術者たちは、世界一高度で優秀であるという評判があるから、大丈夫だと思っている。」

意思決定者 X 「たとえ、クラウド事業者 A の技術者たちは世界一高度で優秀であるとしても、過失や故意による欠陥 (= 不祥事) が発生し得る。どのように担保しているのか。」

技術者 Y 「公的なセキュリティ規格に準拠し、独立主体による監査がなされると聞いている。」

意思決定者 X 「その監査主体の技術者たちは、クラウド事業事業者 A の世界一高度で優秀な技術者たちと同等程度に優秀であるのか。本当に、クラウド事業者 A の Trusted Zone を保護するソフトウェア・コードに関して、開発したプログラマたち以上の能力でこれを監査できるのか。」

技術者 Y 「それは、当然、困難であろう。クラウド事業事業者 A の世界一高度で優秀な技術者たちの技術レベルは、創造性を発揮できる人たちばかりの水準だ。監査会社にはそのようなレベルの人はあまり集まらず、能力がどうしても不足する。」

意思決定者 X 「そうすると、監査主体は、クラウド事業者 A の通常の監査対象オペレーションを監査することはできるが、クラウド事業者 A の高度な技術者を監視・統制する視点での監査には、事実上の能力限界があるということで合っているか。」

技術者 Y 「その通りである。」

意思決定者 X 「それでは、監査主体が能力の不足からクラウド事業者 A の高度な技術者の過失・故意を見落とした場合に、監査主体はわれわれユーザーに対して何らかの責任を負うのか。」

技術者 Y 「監査主体としては、『通常想定されるオペレーション的部分の監査は確実に行なっていた。これを超える高度な技術者の自由裁量範囲まで監査することは社会通念上不可能だ。今回は、その不可能な領域で発生した事故である。よって、監査主体としては、結果回避可能性がなかったから、責任はない。』と反論し、免責を主張するであろう。」

意思決定者 X「すなわち、クラウド事業者 A の基盤サービスの Trusted Zone とされる領域にセキュリティ上の欠陥があり、監査主体がこれを見落としていたとして、われわれユーザーに損害が生じても、監査主体は、われわれユーザーに対して、監査能力不足の限界が原因で生じた責任を賠償しないということか。」

技術者 Y「そのとおりである。それにまた、監査主体とわれわれユーザーとの間では直接の契約関係がなく、不法行為責任しか追及できないが、これには不法行為の証拠を得る必要があり、そのような証拠収集は事実上困難である。」

意思決定者 X「それでは、監査があることは Trusted Zone のセキュリティ侵害の原因となるクラウド事業者 A の高度な技術者の故意・過失の発生を全然予防できていないことにならないか。」

技術者 Y「まあ、実際は、そのとおりである。そこまで気付いているユーザー側経営者は少ないが、賢明な経営者は気付いている。ただ、技術者たちはこのようなリスクを、経営者が気付かない間は、わざわざ経営者に話さないようにしている。仕事が増えるためである。だが、あなたは経営者でありながら、随分と技術の本質を突いてくるので、ついうっかりと、話してしまった。」

意思決定者 X「クラウド事業者 A のサービスは、多数のユーザーが利用しているのか。」

技術者 Y「その通りである。数百万ユーザーが利用しているとされる。」

意思決定者 X「数百万ユーザーをそれぞれ狙う各々の攻撃者の中には、クラウド事業者 A の世界一高度で優秀な技術者たちの技術レベルと同等以上の技術レベルを有する攻撃者が存在するか。」

技術者 Y「それはもちろん、確実に存在する。数百万ユーザーの中には、政府などの行政省庁、大企業などの莫大な情報資産を、クラウド事業者 A の Trusted Zone を信じて無防備に預けているユーザーが多数存在する。これらを狙う者が雇用する攻撃者のレベルは、予算に糸目をつけないから、極めて高いレベルを有している。」

意思決定者 X「そのような高いレベルの攻撃者が、あるユーザーを狙って、クラウド事業者 A の Trusted Zone に対する攻撃を成功させたとき、横展開して、

それ以外のユーザーにも影響が生じるリスクはあるか。」

技術者 Y 「そのリスクは、極めて高い。横展開というよりも、瞬時の同時的展開が可能というべきである。ひとたびあるユーザー 1 のための Trusted Zone を統制する部分を掌握した攻撃者は、その部分がユーザー 2, 3, 4, ... と共通のソフトウェアで動作しているだけでなく、実際の基盤としても共通で、これが論理分割されているということにすぐさま気付くであろう。貸金庫室に侵入したら、ターゲット以外の金庫の財宝も手に入るという具合である。これは、従来よりも比較にならないほど、危険が大きい。従来であれば、大規模に利用されているソフトウェアに瑕疵があっても、これが物理的・論理的にさまざまな多様な方法で、伝統的オンプレミス環境で、ファイアウォール等で多層防御されていた。金庫はそれぞれの人が持っていたのである。1 の攻撃者は、同時に 1 の組織にしか侵入できなかった。ユーザー間の論理境界は、物理法則と、多様性によって、十分に隔離されていた。だが、単一の自社開発ソフトウェアを単一のパブリッククラウド事業者が、数百万のユーザーのために論理分割して、貸金庫室のような Trusted Zone を作り出したとき、攻撃者が、いよいよ、この論理境界を破ることは、伝統的システムの各ユーザー間の境界を破ることよりも容易であり、一瞬で達成され得る。」

意思決定者 X 「ところで、クラウド事業者 A は、有限責任の法人か。」

技術者 Y 「その通りである。株式会社である。何か関係があるのか。」

意思決定者 X 「クラウド事業者 A には、信用資力はあるのか。」

技術者 Y「公開されている財務諸表によると、それなりに資産があるとされる。」

意思決定者 X 「クラウド事業者 A は、株式会社とのことであるが、それらの資産のうち、株主に配当しようとすればすぐにでも当配できる限度額と、配当が規制されている額との割合はどうか。」

技術者 Y 「資産のうち、株主にいつでも配当してしまえる部分は結構多い。株主たちは、雲行きが怪しくなったら、急いで配当を決議し、お金をその株式会社から引き上げそうである。」

意思決定者 X 「それでは、クラウド事業者 A において、数百万ユーザーが利用している Trusted Zone の部分のソフトウェア欠陥が原因で、数百万ユーザー

同時に損害が発生したとき、すべてのユーザーが賠償を求めるが、それを支払える信用資力は、クラウド事業者 A にはないということか。」

技術者 Y 「その通りである。セキュリティ被害を受けたユーザーたちに本来支払うべき賠償金のほうが、クラウド事業者 A の信用資力を上回る。」

意思決定者 X 「それでは、クラウド事業者 A は無資力となって破たんしてしまうのではないか。」

技術者 Y 「大丈夫である。クラウド事業者 A と各ユーザーとの間には、契約約款があり、そこには、賠償額は直近 1 年で支払った料金と同額を上限する、というような損害賠償の上限が記載されている。だから、大して賠償はしなくてよい。クラウド事業者 A は倒産することなく、安全である。」

意思決定者 X 「その犠牲は何か。」

技術者 Y 「われわれユーザー側が受ける損害が賠償されないことである。」

意思決定者 X 「それでは、結局、『われわれユーザーとクラウド事業者 A との責任分界モデルにより、Trusted Zone のセキュリティの責任はクラウド事業者 A が負うことになっている』という責任負担は現実とは異なり、現実には、Trusted Zone の部分の責任もユーザー側が背負っていることにならないか。」

技術者 Y 「いってしまえば、そうである。(a) クラウド事業者とユーザーとの間で、表見上責任が互いに分担されていて、クラウド事業者は彼らの料金の支払を受ける。(b) しかし、実際に事が起こった場合の賠償責任は果たされない。このような表見上重い責任を負っているように見せて、技術上・実際上は十分な責任が果たされていない、という、経営者から見たリスクを軽く見せ、本当のところの技術者からみた実際上の重いリスクとのギャップで利益を得るというのが、クラウド事業者のビジネス・モデルである。」

意思決定者 X 「それでは、クラウド事業者 A の Trusted Zone の安全性に依存することは危険である。」

技術者 Y 「われわれは大口ユーザーであるから、クラウド事業者 A の役員とミーティングをする機会があった。クラウド事業者 A の役員とミーティングをしたが、『信頼してほしい』、『私が保証する』、などと言っていた。だから、大丈夫だ

と思う。」

意思決定者 X 「その役員個人が『私が保証する』と言ったとのことだが、保証は書面でなされ、かつ、極度額が明記されているのか。」

技術者 Y 「いや、口頭のみである。」

意思決定者 X 「それならば、その保証の約束は、無効 (何ら約束していないことと同じ) となってしまい、保証はないということになる (民法 446 条 2 項)。仮に保証があったとしても、その個人の資力では足りない可能性はあるのだが。それでも、随分心強いことである。『私が保証する』と言ったその保証の約束を、その役員に、明確に、書面に書いてもらうことはできないのか。」

技術者 Y 「実はそう思って、交渉してみたら、その役員も、実は Trusted Zone の安全性を自分で検証した訳ではなく、安全かどうか分からないから、やっぱり保証はできないと言い始めて、ついに、『私が保証する』という言葉を、撤回してしまった。」

意思決定者 X 「そうすると、担保は何もない、ということか。」

技術者 Y 「そのとおりである。」

意思決定者 X 「それでは、クラウド事業者 A の Trusted Zone を用いるサービスは利用すべきではないと言わざるを得ないが、それでも、利用したいのか。」

技術者 Y 「技術的には、楽なので、是非利用したい。確かに上記の議論を経た場合、完全性・機密性・可用性の点で、セキュリティ上問題があると認めるしかない。しかし、普段は、性能が良く便利であるためである。」

意思決定者 X 「それでは、どのようにすればよいのか。」

技術者 Y 「クラウド事業者 A の Trusted Zone の部分について、利用する際には、次の 3 点に注意しようと思う。

(a) データを保管する際には、われわれユーザーの側でも、暗号化を施す。クラウド事業者 A に瑕疵があったとき、他のユーザーのデータは奪取されるが、われわれのデータは無事である。

(b) データは必ずローカルで、または完全に異なる系統の他のクラウド事業者のストレージで、頻繁にバックアップする。先の議論を踏まえると、単にリージョン

が異なるというのではなく、異なる事業者のクラウド基盤である必要がある。クラウドのデータ保管の仕組みはとても複雑なプログラムコードで動作していて、メタデータ破損だけでデータ本体が全く取り出せなくなるリスクがあるが、これを回避する。そして、あるパブリッククラウド事業者が障害で長期間停止したならば、バックアップデータを用いて、他のパブリッククラウド事業者と契約して、またはオンプレミスで、すぐに同等のシステムを継続運用できるようにしておく。これができないような固有の機能は、研究開発目的をのぞき、決して利用しない。

(c) データの破損可能性を考えて、ひんぱんに格納データの整合性を検査する。クラウド API 側が表示するハッシュ値の比較では不十分である。これはデータが保存された瞬間のハッシュ値であり、データが変質しているかも知れないためである。クラウド事業者は、『定期的に』全データをスキャンしてハッシュ値と突合していると言っているが、それにはコストがかかるから、彼らはそれを実際には行なっていない可能性が想定される。こういう検査の手抜き事例は、IT 業界以外には山ほどある。IT 業界のみ手抜きがあり得ないという合理的な理由はないためである。」

意思決定者 X 「あなたが今述べた要素についていまいちど考えると、(a) は機密性、(b) は可用性、(c) は完全性の話であろう。それでは、これらの措置がなされていれば、クラウド事業者 A の Trusted Zone の部分の安全性が崩れても、われわれの情報セキュリティの安全性は技術的に維持されるのか。」

技術者 Y 「技術的に維持される。クラウド事業者 A の Trusted Zone を本質的に Untrusted Zone と同等とみなして、その上に自ら Trusted Zone を実現するという、ゼロトラストセキュリティを実現しているためである。」

意思決定者 X 「(a) のユーザー側での暗号化には、コストはかからないのか。」

技術者 Y 「コストはほとんど増えない。最近の CPU は十分高速であり、数 Gbps の暗号化処理速度が出る。十分検証された安全な暗号化ライブラリは、無料で利用できる。」

意思決定者 X 「(b) のバックアップには、コストはかからないのか。」

技術者 Y 「バックアップにはコストがかかるが、発生し得るデータ喪失の損害

と比較して微々たるものである。」

意思決定者 X 「(c) の検証には、コストはかからないのか。」

技術者 Y 「データを一度読み出して、スキャンし、ハッシュ値を計算する処理には、コストがかかるが、バックアップのコストよりもさらに少なく、発生し得るデータの気付かない間の破損による将来の大損害と比較して微々たるものである。」

意思決定者 X 「(a), (b), (c) を施すと、パブリッククラウド事業者 A への支払額は増えるのか。」

技術者 Y 「若干 CPU 処理時間が増え、バックアップと読み出しのデータ転送時間が増えるが、微々たるものである。」

意思決定者 X 「了解した。それらの方法であれば、ゼロトラストセキュリティの原則に基づいて、パブリッククラウド A を利用しても良さそうである。このゼロトラストセキュリティ原則に基づくパブリッククラウドの利用は、誰も損をせず、セキュリティを確保でき、非常に良い利用手法である。ユーザー側としても良いし、パブリッククラウド事業者としても、若干 CPU 処理時間、データ転送量で受け取る利用料金額が増え、かつ、ユーザー側での対策により、ユーザーから莫大な損害賠償請求を受けるリスクも減り、皆利益を受ける。

ところで、なぜ、このような良いパブリッククラウド利用手法があまり体型化されておらず、この国のユーザー組織の間に周知されていないのか。われわれは本日の議論で幸運にも防護の方法に気付いて良かったが、他の組織は、皆無防備のほうに見える。」

技術者 Y 「それは、経営者であるあなたが随分とさまざまな点を追求してきたからである。普通は経営者はそこまで考えない。技術者としては、表見的セキュリティの範囲内で適当に応答していれば楽である。だが、経営者が技術を学ぶと、こういった本質的議論がなされる。それは短期的には技術者としてみると仕事が増えるので大変だが、長期的にみると、事故のリスクがなく安心でき、また事業継続性に資するので、自らの身のためにもなり、とても良いことである。このような考え方により、経営者と IT 技術者との間の共同の利益が、いよいよ、実現されるのである。」

(2) パブリッククラウド環境において、数百万社の企業が同じクラウド基盤システムを利用していれば、脆弱性を発見され攻撃される危険がはね上がる

従来型のオンプレミスシステムというものは、住民が所有権を有している戸建の建物のようなものである。これが集約されると、マンションになる。マンションにも、区分所有と、賃貸とがある。クラウドは、賃貸である。住人には、所有権がない。賃借権という、とても弱い債権しかない。これでは、とても安心して定住できない。そこで、現実世界の賃貸マンションは、「借地借家法」で規制されるようになっている。この法律によって、容易に値上げはできないから、安心して定住できる。他方、ウイークリー賃貸マンションというものが存在する。ウイークリー賃貸マンションは、ホテルと同様に、「旅館業法」が適用される。住民は安心して定住することはできない。

パブリッククラウドサービスは、マンションと類似である。そして、もともとは、パブリッククラウドサービスは、一時的・短時間に住む人のために作られたものである。Web アプリ等のサービスを定住的に構築するために、初期投資を抑えるため、一時的にクラウド上で大規模にスケールできる従量課金型のサーバー資源があれば、スタートアップ企業にとって便利である。このように、パブリッククラウドは、一時利用・短期間の居住を前提として作られているのに、いつの間にか、意外にも、パブリッククラウドがとても楽だということで、ここに IT システムを本格的に構築し、定住するユーザーたちが出現した。パブリッククラウド事業者も、そのような定住ユーザーへの売り込みを本格化した。

ところが、パブリッククラウドサービスの分野は、法的規制が未発達であるため、未だに、借地借家法のような保護がない。ウイークリーマンションと同様、住民はほとんど保護されない。クラウド事業者は、いつでも大幅な一方的条件変更ができてしまう。だから、本来、ユーザーは安心して定住できない。しかし、一度ウイークリー賃貸マンションや漫画喫茶のようなもので生活することに安心し始めたユーザーたちは、自らの住居を維持・管理する面倒さも相まって、パブリッククラウドに定住してしまったのである。

通常のパブリッククラウドは、通常の規模のウイークリー賃貸マンションのよう

なものである。さまざまなマンション賃貸業者が存在し、さまざまなマンション管理人が分散して管理している。それぞれのマンションの構造は、互いに異なる。脆弱性も異なる。攻撃者は、多数のマンションに同時に攻撃を加えることはできない。1 つのマンションに攻撃を加えるためには、そのマンションに関する研究が必要となり、コストがかかるためである。このような、多数の戸建や小規模なマンションで構成される多様性と分散性により、全体的に見ると、極めて高いセキュリティが実現される。ただし、この場合、各戸建やマンションの住人は、それぞれ、最小限度のセキュリティ・リテラシや家屋修繕の知識を有する必要がある。そうしないと、小規模な被害であるが、空き巣被害が確率的に発生し得る。また、分散的に故障が発生し、雨漏りを防ぐために、時々、修繕が必要となる。ただ、これらは分散的に発生する。全戸で同時に空き巣被害や故障が生じることはない。

これと比較して、数百万社が利用するような大規模なパブリッククラウド事業者のクラウド基盤は、数百万人が住んでいる 1 棟の超高層タワーマンションのようなものである。確かに、マンション入口にオートロックのシステムがあり、各戸にも警備システムがあり、住民は自らセキュリティ意識を持たなくとも、それなりに安全に暮らせる。ところが、クラウド基盤に欠陥があった場合に生じる完全性・機密性・可用性の問題は、数百万人すべてに、同時に影響する。全員が、瞬時に同時に被害を受ける。高度なサイバー攻撃者は、基盤部分への攻撃が可能である。超高層タワーマンションの管理人室にすべての戸室のカギが置いてあり、すべての戸室のセキュリティシステムに通じた配線が通っているようなものである。そして、意外にも、管理人室は防備が薄かったりする。いったん管理人室を掌握した泥棒的攻撃者は、すべての戸から財物を窃取できる。すべての戸の換気システム、給電システム、給水システムがつながっており、これらの共通システムに故障が生じた場合も、テロ攻撃者による攻撃が加えられた場合も、すべての住人に被害が及ぶ。1 人の住人を狙って加えられる攻撃により、すべての住民が巻き添え被害を受けることになる。

パブリッククラウドを超高層タワーマンションにたとえるとき、「単一パブリッククラウド事業者のサービスであっても、リージョンで分離されているから、リー

ジョン間をまたがって利用すれば、一極集中の危険は高くない。」という考え方が生じる場合がある。確かに、リージョン分離が完全になされていて、各リージョンをまたがって分散的に利用しているならば、一見、複数のクラウドサービス事業者を分散して利用しているように錯覚し、安全なように誤信する。しかし、実際にはその方式では安全性は担保できない。その理由は、次の 3 つである。

第一に、単一のパブリッククラウド事業者が複数のリージョンを構築したとしても、各リージョンのセキュリティ領域は、物理的に隔離されているだけで、論理的には隔離されていない。各リージョンのセキュリティ領域には主要な共通的部分が存在する。その証拠に、ユーザーの視点においても、クラウドサービス事業者の管理画面を見れば、すべてのリージョンのリソースに単一の管理特権ログインで分け隔て無くアクセスできて、任意の管理が可能で、これはとても便利である。ユーザーの視点でこれらが可能であるということは、サイバー攻撃者の視点、すなわちクラウドサービス事業者の特権制御機構を掌握した攻撃者の視点では、同様に、リージョン間の隔離は無意味なものとなる。マンションにたとえてみれば、同一系列の超高層タワーマンションが数棟立っていて、外からみると独立しているように見えるが、地下にインフラ地下道 (共同溝) があり、換気ダクトや制御システム、管理人の特権的メンテナンス用通路、スペアキーを保管している管理室の金庫などが同一になっているとか、見かけ上別々でも同一の制御でこれらがすべて乗っ取ることができるというような状態になっているという具合である。

第二に、単一のパブリッククラウド事業者の各リージョンの基盤ソフトウェアは、同一である。パブリッククラウド事業者が異なれば、基盤ソフトウェアが異なり、多様性が存在する。これと比較して、パブリッククラウド事業者 1 社の複数リージョンを利用している場合、ソフトウェアの脆弱性は同一のものが共通している。攻撃者は、1 つの攻撃手法の発見により、すべてのリージョンを同時に攻撃できる。マンションの例えでいうと、たとえば 3 棟のマンションがすべて同一の建築方法で建立されており、特定の種類の同じ振動を与えると倒壊してしまうとか、特定の種類の裏口の脆弱性が存在するという具合である。

第三に、パブリッククラウド事業者そのものの信用リスクの隔離がなされていな

い。パブリッククラウドを利用する上での最大のリスクは、クラウド・ショックの発生である。クラウド・ショックの発生時においては、多数のユーザーが、信用不安が生じたパブリッククラウド事業者からデータを一斉に引き出そうとする。その信用不安の範囲は、パブリッククラウド事業者単位で発生する。リージョンは無関係である。したがって、すべてのリージョンで、同時に、データ渋滞が発生してしまう。このように、リージョン分離は、パブリッククラウド事業者のソフトウェアの瑕疵からのユーザー保護のためには、十分に役に立たない。リージョン分離は、データセンタの火災や外部とのネットワークの故障の際に役立つ程度である。よって、単一のパブリッククラウド事業者を用いる限り、リージョンが分かれていても、単一でも、パブリッククラウド事業者起因のソフトウェアの欠陥によって生じるユーザーが被る損害のリスクは、ほとんど変わらない。

パブリッククラウドにおいては、1 つのクラウド事業者のクラウドサービスのユーザー数が多ければ多いほど、上記の一極集中のセキュリティ上の危険が増大する。マンションの例で考えると、構造や共有部分のセキュリティのリスクが増大すると、損害賠償リスクの金額が高まる。一見、オンプレミス (戸建の所有) よりもパブリッククラウド (ウイークリーマンションの賃貸利用) のほうが便利であり、リスクも低いように見える。小規模なパブリッククラウド事業者であれば、確かにリスクは低い。ところが、極めて大規模なパブリッククラウド事業者には、極めて大きなリスクが潜む。そのリスクは、10 年、20 年に 1 回程度しか現実化しないかもしれないが、現実化した場合は、同一の大規模パブリッククラウドに居住するすべての住人に対して、一度に、甚大な被害を生じさせる。これは、パブリッククラウド事業者の過失または故意によって引き起こされる。住人の責任ではない。だから、住人たちは、個別に、パブリッククラウド事業者にその損害を賠償請求することになる。ところが、それが発生すると、パブリッククラウド事業者は支払えないほどの巨額の賠償義務を抱えて、事業破綻してしまう。そこで、これを防ぐために、パブリッククラウド事業者は、予め利用規約上に損害賠償金額の上限額を規定している。たいていは、直近 1、2 年間にユーザーが支払った料金額を上限し、特別損害は一切補償しないというような内容になっている。これにより、パブリック

クラウド事業者は、直近 1、2 年間に受け取ったお金だけを残しておけばよいことになる。残りは株主配当や投資 (すなわち、新たな利益を生み出す活動) に回してしまっても、それ以上の損害賠償は支払わなくて良いので、理論上は、大事故が起きても、倒産しないことになる。これによってパブリッククラウド事業者が破たんするリスクは解決されているから、仮にパブリッククラウドの基盤の重大なセキュリティホール (それは、必ず存在する。) が攻撃者によって突かれて、すべてのユーザーが被害を受けても、パブリッククラウド事業者が賠償責任によって破たんするリスクはなく、「大いに安心である」、という理論が存在する。近年見られる誠に奇妙な現象は、パブリッククラウド利用に関するリスクを議論しようとすると、この理論をユーザー側 (特にユーザー側技術者) が掲げて、「この損害賠償金額の上限の規定のおかげで、大事故があっても、クラウド事業者が破たんすることがないから、安心である。」という風に説明をして安心を得ようとすることが散見される点にある。これは、誠におかしな理論である。われわれユーザー側は、そのような事態が発生した場合は、パブリッククラウド事業者の過失責任を追及し、賠償責任を果たしてもらわなければならない。ところが、契約約款において上限が規定されていると、それ以上の被害はユーザー側の泣き寝入りということになる (もちろん、重過失または故意を証明できれば、裁判所によって、約款上の上限規定は無効となるであろう。しかし、その証明は容易ではない。)。 そうすると、ユーザー側としては、全く「安心である。」という考えには至らないはずである。ユーザーとパブリッククラウド事業者とは、この点において、利益相反関係にある。ところが、ユーザー側技術者は、なぜか、パブリッククラウド事業者の立場に立って「安心である」と発言してしてしまうのである。技術者は、パブリッククラウドが技術的に好きだから、そのような立場の混同をしばしば行なう。経営者は、それをよく見抜いて、パブリッククラウドを利用する際のリスクを正しく認識しなければならない。

パブリッククラウド事業者のユーザーは、原則として、パブリッククラウド事業者の利用規約に記載されている賠償上限 (この賠償上限額の規定が、SLA = Service Level Agreement の最重要の要素である。) を想定し、パブリッククラウ

ド事業者の基盤システムのセキュリティ (パブリッククラウド事業者の内部の Trusted Zone そのもののセキュリティ) が破られた場合において、利用規約に記載されている賠償上限以上の損害が発生し得る情報やシステムを、決してパブリッククラウドに置いて運用してはならない。ただし、ユーザー側におけるクライアントサイドの暗号化を確実になす場合 (そして、パブリッククラウド上に決してその秘密鍵を一時的にしろ置かない場合) に限り、機密性喪失による損害額が利用規約上の上限を超える情報をパブリッククラウドに置くことは、許容される。また、マルチクラウドやオンプレミスを活用し、すべての必要なデータの全ローカルバックアップを常に確保し、いざという時に代替システムを直ちに稼働できる状態にしておけば、パブリッククラウド上で重要なシステムを稼働させることも、許容される。あるいは、パブリッククラウド事業者から確実な担保の差し出しがあり、パブリッククラウド事業者のシステムセキュリティに欠陥があった場合の損害額は、裁判所によって認定された損害すべてが、規約上の上限額なく、すべてその担保から支払われるような保全がなされているのであれば、前記のような暗号化や分散、ローカルバックアップなしで、すべての機密情報を単一のパブリッククラウド事業者に預け、すべての IT システムを当該パブリッククラウド事業者に委ねても、安心である (パブリッククラウド事業者に瑕疵があった場合の損害は、その担保から得れば良いためである)。だが、こういった対策をとらずに、パブリッククラウド事業者のセキュリティを一方的に信用し、クライアント側暗号化も、分散も、ローカルバックアップも、速やかに代替システムをローカルで動作させることができる体制も整えずに、パブリッククラウド事業者にすべてを委ねるのは、経営上の任務懈怠とみなされるおそれがある。パブリッククラウド事業者が SLA 契約上において賠償責任の上限額を規定しているという事実は、パブリッククラウド事業者自らが、「当社には、当社のシステムに瑕疵が合った場合でも、SLA で規定している賠償金額以上の問題を防止するセキュリティ上の保証は、提供できません。現在の世界の技術水準で実現可能なセキュリティの水準は、その程度であるためです。」というメッセージを、ユーザーに対して発していることと同義である。そして、パブリッククラウド事業者のセキュリティ欠陥によってユーザーシステムのセキュリティが

侵害されるリスクは、容易に想定できる。経営者の視点でみると、「予見可能性」があるということになる。これを予防するための、クライアント側暗号化やクラウド分散、ローカルでの代替システムを直ちに稼働できる状況の整備といった対策も、低コストで、効果的に実施することができる。これらを妨げる要因は存在しない。経営者の視点でみると、「結果回避可能性」があるということになる。本日時点で、パブリッククラウド事業者の事故によって発生する甚大な被害に関しては、予見可能性があり、結果回避可能性がある以上、これを怠った経営者は、いざパブリッククラウド依存によって事故が発生し、パブリッククラウド事業者が約款上の補償上限として規定している金額以上に損害が発生した場合、民間企業であれば、その経営者個人は、任務懈怠責任を追及され得る。官公庁であれば、その意思決定管理者は、重過失による国または地方公共団体からの求償権を行使され、個人財産が危険にさらされる。したがって、われわれは、この問題を、他人事として考えることはできない。われわれは、パブリッククラウド依存への危険性を認識し、また、その危険を減少するための方法が用意に作為可能であることを認識している以上、できる限り、損害リスクを軽減するための議論を行ない、対策を行なう法的義務を有しているということができるのである。

また、法的責任の問題は別においても、われわれは、この「国・地方ネットワークの将来像及び実現シナリオに関する検討会」という公式な会議において、議論をして意思決定をする以上、明らかに日本国の主権者に対する社会的責任・同義的責任を有している。われわれの上位層である政治家の方々は、国民の投票選挙で選ばれた立派な方々である。政治家の方々は、国民の絶大な信任を得ている。その政治家の方々から、国・地方ネットワークに関する重要な問題についての技術的・仕組み的検討が、われわれ「国・地方ネットワークの将来像及び実現シナリオに関する検討会」に、今、この瞬間、託されている。すなわち、国民の方々は、日本の代表者である政治家の方々は IT 技術やサイバーセキュリティに関してとても高い能力を有しているからということで、信頼して、政治家の方々に票を入れ、これによって選出された政治家の方々は、必ずしも細部に関する事柄については専門家に任せたほうがよいということで、われわれ「国・地方ネットワークの将来像及び実現

シナリオに関する検討会」にその検討を全面的に任せているのである。そうすると、国民たちの信任は、政治家を経由して、間接的に、われわれ「国・地方ネットワークの将来像及び実現シナリオに関する検討会」が、今の瞬間は、担っているということができる。「国・地方ネットワークの将来像及び実現シナリオに関する検討会」は、この重要な問題について、日本国という極めて大きな船を運航するために設置されている操舵席における舵の方向を絶妙にコントロールする仕事を委ねられた合議型の集団船長のようなものである。われわれの舵取りにおいて、少しでも気を抜くと、前方に存在するかも知れない水面下の大きな障害物や、よく見ると視認可能な氷山を、発見することができなくなる。そういったものを予見して検出可能であったにもかかわらず、これを無視すれば、それはわれわれの責任である。そして、もっとも良くないことは、予見していても、そのことを誰も指摘しないことである。そのような不作為は、故意との同価値性を有するのであり、その責任は、極めて重い。われわれの後継者たちはわれわれの選択を承継する。われわれの現代の決定においてもはや後戻りすることができない地点を通過する可能性がある。この場合、後継者たちには結果回避可能性がないから、後継者たちには事故の責任はない。その責任は後継者たちでなく、われわれにある。

　われわれは、「国・地方ネットワークの将来像及び実現シナリオに関する検討会」の重要な意思決定者集団である。将来、パブリッククラウドの欠陥が原因で、いざ大事故が発生し、国や自治体のシステムが止まったり、膨大な情報の機密性が失われたりしたというような場合、それは、日本における歴史的な著名な大イベントとなってしまい、もしかすると、日本の歴史書に載るであろう。パブリッククラウドに関する危険性を認識・容認していながら、このことを議論の俎上に載せずに、安易に意思決定をしてしまったということが、将来の事故の被害を受けることとなる世代に知られると、その原因は、あの 2023 年の「国・地方ネットワークの将来像及び実現シナリオに関する検討会」での検討が不十分であったことにある、というように後世に強調的に指摘され、われわれ個人が、それぞれ、上記のような法的責任や社会的責任を、あたかも戦犯のように、追及され得る。それは、避けたいのである。結果が名誉なこととして歴史書に載るのであればそれは結構なことである

が、不名誉なことでその結果が歴史書に載ることは是非とも避けたいのである。

(3) パブリッククラウドの事業者内秘匿基盤ソフトウェアの脆弱性は、ユーザーがバイナリすら見ることができず、衆人環視が困難であることから、当該基盤の上は原則としてゼロトラスト原則上の Untrusted Zone として扱い、その上に暗号化等の論理的手法を加えることにより Trusted Zone を構築する必要がある

パブリッククラウド基盤上にさまざまなサーバーや Web アプリケーションやストレージやデータベースを構築する場合、それらの領域を作り出すパブリッククラウド事業者の基盤ソフトウェアによって構成される仮想的なプラットフォームの上で構成される世界は、原則として、すべてゼロトラスト原則における Untrusted Zone としてみなす必要がある。その理由は、現在の技術的水準の限界により、パブリッククラウド基盤を構成する各種システムソフトウェアは、欠陥が存在する可能性が極めて高く、従来の IT システムを構成する OS やネットワーク機器、ファイアウォール等のソフトウェアと比較して比較にならないほど危険であるためである。このことについて、もう少し詳しく検討しよう。

ゼロトラスト原則は、Trusted Zone とみなす領域を極小化し、自らセキュリティを確実に支配・管理できる領域のみを Trusted Zone として特権領域化した上で、Trusted Zone 以外のすべての場所・主体・要素 (Untrusted Zone) からのアクセスを認証・認可・監査するという原則である。Trusted Zone を極小化することにより、ユーザー組織は、Trusted Zone のセキュリティを高めることに集中できる。ゼロトラスト原則においては、社内 LAN を含めた大半の領域を Untrusted Zone とみなし、Untrusted Zone をインターネット等の公衆網と同等に扱う。したがって、社内 LAN 等の物理的ネットワークに対する完璧な防備は不要となる。その反面、Trusted Zone においては、従来と比較して、極めて高いセキュリティ水準を必要とする。ユーザー組織自ら、またはユーザー組織が直接信頼する極めて高いセキュリティ能力を有する者 (たとえば、十分なセキュリティ能力を身に付けた従業員や委託先) によって、そのように作られる Trusted Zone のセキュリティ水準を、自らの手で、十分に検証しなければならないことになる。従来のインフラにおいては、物理的な面での隔離が十分であるかどうかを検証すれば良かった。

たとえば、Trusted Zone が特定の極めて閉鎖的な LAN によって実装されている場合で、その閉鎖的 LAN が単一のサーバールーム中の固有のスイッチによって閉域化されている場合、その閉鎖的 LAN が他と物理的に接続されていないこと、サーバールームの室への物理的侵入の困難さが社会的相当性の程度を満たしていることの 2 点を検証すれば良かった。もちろん、Untrusted Zone と Trusted Zone との境界にあたる認証機能付きサーバーのセキュリティの維持は、厳格になされる必要があるが、これは Untrusted Zone と Trusted Zone との分離があってもなくても必要なことであり、この議論においては、本質的事柄ではない。さて、今日の IT システムにおいては、Trusted Zone の Untrusted Zone からの隔離は、物理的な隔離ではなく、ソフトウェア仮想化技術によって構成される。社内においては VM 技術、仮想スイッチ技術、L2 スイッチの VLAN 技術、ファイアウォール等を用いて Trusted Zone を最小化する。VM や仮想スイッチ、VLAN 等の技術は、IT の標準的用語概念でいう「枯れた技術」であり、長年極めて多数のユーザーによって検証されてきて、さまざまな脆弱性が指摘され、修正されてきた。これらの仮想的隔離技術は、ユーザーの手元にその実装としてのソフトウェアバイナリやファームウェアが提供される。セキュリティ能力を一定水準身に付けたユーザーが、世界中の多くの組織に存在する。これらのユーザーは、手元にあるこれらの仮想化製品やネットワーク製品への攻撃を加える。その際に、ソフトウェアやファームウェアをリバースエンジニアリングする。ソースコードがあればそれを読むが、ソースコードがなくとも、逆アセンブルすれば、脆弱性に気付くことができる。脆弱性に気付いたら、発見者は開発元に対して、それを修正するよう勧告し、修正されるか、または修正がいつまでもされなければ、その脆弱性情報を Web 上で公開する。このように、従来のプロプライエタリな仮想化技術製品は、極めて多数のユーザーによる衆人環視がなされており、脆弱性の芽は、開発元のエンジニアではなく、外部のエンジニアによって、摘み取られてきたのである。このような仕組みは、過去 30 年間くらいに渡って機能してきた。これにより、当初は脆弱性が多数あり危険であった Windows の OS のコアのセキュリティ部分、Cisco 社のネットワークファームウェア (IOS と呼ばれる)、VMware 等の著名な VM 製品、各

社のデータベース製品やファイアウォール製品等の主要かつ重大な脆弱性が修正されていき、これらの基盤は今日ではかなり安全な部類のものとみなされるようになった。たとえば、20 年くらい前までは、企業システムにおいて Windows を使用することは、勇気があることであるとみなされた。今日では、Windows はかなり安全な OS として認知されており、企業システムで Windows を利用することは一般的となった。それは、前記のように、Windows の実物バイナリを、世界中の多数の技術者、研究者たちが、リバースエンジニアリングし、さまざまなサイバー攻撃的試行を自らの実験室内加えて、脆弱性を発見し、これを Microsoft 社に改修するようつとめて勧告してきたためである。Microsoft 社の Windows の重大かつ主要な脆弱性のリストをみると、たいてい、外部のエンジニアによる発見であることを示す謝辞の記載がある。

このように、現在の技術水準では、あるシステムソフトウェアのセキュリティ上の欠陥を発見し修正するための方法は、外部の多数の能力の高いユーザーたちに、セキュリティ上の欠陥を発見してもらい、その情報提供を受けて修正するという方法しか、存在しない。そして、ソフトウェアの規模や複雑さに応じて、一定の割合で脆弱性が存在する。したがって、あるソフトウェアが安全か否かは、そのソフトウェアが、どれだけ多数の合計時間をかけて、多数のユーザーによって衆人環視されて検証されたかに比例する。衆人環視によるソフトウェア分析が十分になされているソフトウェアは、安全性が極めて高い。衆人環視によるソフトウェア分析が全くなされていないソフトウェアは、安全性が極めて低い。

だが、このパブリッククラウドのセキュリティ問題は、やがて、ゆるやかに解決されていくであろう。いずれ、パブリッククラウド基盤を構成するシステムソフトウェアも安全なものになるであろう。これは、次のような状況の改善によってなされる。現状、これらのソフトウェアは、各パブリッククラウド事業者でそれぞれ別々に開発されている。これらのソフトウェアは、各パブリッククラウド事業者のデータセンタの内部の彼らの所有サーバーでのみ実行され、プログラムのバイナリを含めて門外不出とされている。これは、パブリッククラウド事業者ができるだけ長い時間、パブリッククラウド領域における健全な競争の発生を回避して、値段を

高止まりさせるためである。すなわち、パブリッククラウド基盤を構成するソフトウェアを作る技術が未だ世の中の人々によって習得が困難である間は、パブリッククラウド基盤を構成するソフトウェアを非公開の状態にしておいたほうが、パブリッククラウドを運営することによって得られる利益は大きくなる。これらのソフトウェアや技術的ノウハウが世の中で普及してゆくと、誰でも大規模なパブリッククラウド事業を立ち上げることができるようになり、先行者利益は失われてしまう。健全な競争が発生し、ユーザーは利益を受けるが、その一方で、パブリッククラウド事業者は利益が減ってしまう。これを回避するためには、パブリッククラウド事業者は、パブリッククラウドを運営するために必須のソフトウェア基盤技術をあえて世の中に公開せず、隠しておいたほうが良いということになる。この点が、従来型ソフトウェアビジネスと、パブリッククラウドソフトウェアビジネスとの根本的な違いである。これを、すべての大手パブリッククラウド事業者が行なっている。これは、彼らの視点では、やむを得ない合理的利益追求行動である。しかし、時間が経てば、誰でも大規模なパブリッククラウドを容易に構築する技術とノウハウが、世の中に解放されて普及されてゆくと考えられる。いくつかの少数の有力なソフトウェア群が広い範囲で利用されるであろう。これらはオープンソースかも知れないし、プロプライエタリかも知れない。いずれでも良い。そうすれば、従来の OS やネットワーク機器、ファイアウォールのように、ソフトウェアの衆人環視が可能となる。たとえバイナリのみであっても、現物がユーザーたちの手元にあれば、誰でもリバースエンジニアリングして、安全性を検証できる。そうすれば、従来の OS 等のように、数多くの脆弱性が発見され、それらが修正されて、安全な状態になる。そのようにして安全性が確保されたパブリッククラウド技術は、現在のように、安全性の衆人環視が全く行なわれていないパブリッククラウド事業者のソフトウェア技術を置き換える。大手パブリッククラウド事業者も、自ら開発した秘匿技術と比較して、安全性が高い公開されているパブリッククラウド技術を使用して、クラウドサービスを運営するようになるであろう。このときになって、はじめて、大手パブリッククラウドサービスは、安全であるということができるようになる。大手パブリッククラウドサービスによって作られる Trusted Zone は、ゼロトラスト

原則上の意味としての Trusted Zone であるとみなしても差し支え無いことになる。しかし、それまでは、まだ数年以上の時間がかかると考えられる。その間は、大手パブリッククラウドサービス事業者を含めたすべてのパブリッククラウド事業者の基盤ソフトウェアは、多くの脆弱性を含んだままの危険な状態に留め置かれる。この領域は、ひとたびサイバー攻撃者によって攻撃がなされると、ユーザーシステムのセキュリティの三大要素である、完全性・機密性・可用性のいずれも侵害されてしまう。ただし、完全性と可用性は、損傷するとユーザーもパブリッククラウド事業者もすぐに気付く。データがおかしくなったり、アクセスできなかったりすると、直ちに苦情を入れるためである。しかし、サイバー攻撃者は、パブリッククラウド事業者のソフトウェアによって構成されている Trusted Zone をソフトウェア欠陥に対する攻撃によって掌握したとき、単一のパブリッククラウド事業者が管理するすべてのユーザーのすべてのデータをほとんど同時に消去することができてしまう。クラウドサービスにおいては、多数のユーザーのデータを効率的に格納するため、ディスクをユーザー間で分離せず、同じディスクに多数のユーザーのデータを混ぜて記録している。そのユーザー間のデータの所有権は、メタデータとして記録されている。実際のディスクから消去をしなくても、メタデータを消去すれば、アクセス消去したのと同じ結果となる。ただし、全部一気に消去すると、一定期間内はバックアップがあるから、そのバックアップがパブリッククラウド事業者の Trusted Zone ではない別の場所に物理的に管理されていれば、何とか戻すこともできる。しかし、攻撃者が時間をかけてメタデータを部分的に損傷していけば、パブリッククラウド事業者は、なかなかこれに気付くことは難しい。メタデータは一般に時系列で世代的バックアップがなされているが、損傷したメタデータのバックアップしか残っていない状態になれば、元のデータを消失させたことと同じ状態に陥る。この状態で実データからメタデータを再構築できる場合がある。あたかもファイルを削除してしまったハードディスクからファイルを復元するのと同様に、極めて長い時間がかかり、不確実性が増大する。その状態では、もはや可用性は失われる。データの取り出しには長時間かかる。このような完全性と可用性に対する侵害よりも、より危険で損害が大きいのは、機密性に対する侵害である。

攻撃者は、10 年以上かけて少しずつ膨大なデータを盗み出すという手法をとることができる。主要なユーザーの機密情報が 10 年かけてわずかずつ漏えいしていたという状況が 10 年後になって初めて発覚する。攻撃者は、パブリッククラウド事業者の CPU 上で暗号化・復号化されるストレージ上のデータの平文にアクセスできる。そのリスクは、ユーザー提供型の共通鍵が都度使用される方式の API であっても、変わらない。パブリッククラウド事業者の Trusted Zone を掌握した状態にいる攻撃者は、ユーザー提供の AES 暗号用の公開鍵の平文にアクセスできるためである。したがって、パブリッククラウドへの機密データ保管においては、クライアント側暗号化が必要不可欠である。だが、パブリッククラウド事業者の特権領域の安全性 (その安全性には、前述の通り、実は十分な担保がない。) を過信していると、クライアント側での暗号化を行なっておらず、平文データを攻撃者に奪取されることになる。これはもちろん、ユーザーとパブリッククラウド事業者との関係では、パブリッククラウド事業者側の責任である。しかし、パブリッククラウド事業者の視点からは、SLA の契約により、直近 1、2 年に支払われた料金を上限とした賠償義務を果たせば良いというように規定されていたりする。パブリッククラウド事業者が賠償リスクによって破たんすることはないので、たしかに、安心である。安心というのは、もちろん、ユーザーにとっての安心ではない。パブリッククラウド事業者の経営者や株主の視点でみて、安心であるという意味である。ユーザーはパブリッククラウド事業者からの被害を弁済されることはなく、他方で、ユーザーの顧客からの賠償請求に対しては全額応じなければならない。なぜならば、パブリッククラウド事業者のソフトウェアが、その秘匿性が原因となり、従来の OS 等のオンプレミス型のソフトウェア基盤と比較して脆弱性が多く存在し得て、サイバー攻撃から脆弱である可能性があるという点は、一般的な水準のユーザーであれば容易に推測することが可能であり (少なくとも、この文章を読んでいるユーザーにおいては、既知の情報となっている)、パブリッククラウド事業者に重要なデータを預ける以上、パブリッククラウド事業者側に瑕疵があった場合に備えて、クライアント側暗号化を施さなければ機密性が危険にさらされるという危険と、これによる損害の発生は、経営者として予測可能であるためである。そして、

クライアント側での暗号化による損害を回避することは極めて容易であるので、結果回避可能性も満たされている。パブリッククラウド事業者のユーザー組織 (国、地方自治体、官公庁、企業) は、顧客 (国民、市民、消費者等) の個人に権利がある情報をこれらの個人の信任を受けて保管しているので、自己の物に対する注意の程度では足りず、善管注意義務の程度に基づいた保管が必要である。したがって、パブリッククラウド事業者にサイバー攻撃者が侵入し、データを盗まれた場合でも、前記の予見可能性や結果回避可能性が存在する以上、個人情報を盗まれた国民、市民、顧客からの法的責任の追及と賠償義務を免れることができない。これを避けるためには、パブリッククラウド事業者に預けるデータには、必ず、クライアント側での暗号化を施す必要がある。

このように、パブリッククラウド事業者によって秘匿されている、衆人環視による手法によって未検証の基盤ソフトウェア群は、検証されていないので、現状では、その上で作られる仮想マシン (VM)、仮想ネットワーク (VPC)、仮想ストレージ、仮想ファイアウォール等は、原則として、Untrusted Zone として扱わなければならない。ただし、例外として、パブリッククラウド事業者との間で SLA の契約があり、かつ、その SLA の契約において、「損害賠償金額の上限は、直近 1 年間の料金額と同額を上限とする。」、「特別損害は補償しない。」等のパブリッククラウド事業者に一方的に有利な免責規定が存在せず、機密性・完全性・可用性の喪失によってユーザーが被ったと裁判所によって認定されるすべての損害をパブリッククラウド事業者が全額保証する契約 (そのような契約が、本来は、普通の契約である。) となっていて、抜け穴的な例外条項が塞がれていれば、従来の OS 等で施されている衆人環視型の脆弱性検査が未了のパブリッククラウド事業者のソフトウェア基盤によって作られるゾーンは、Trusted Zone であるとみなしてもよい。なぜならば、その領域に存在する脆弱性が原因でセキュリティが破られたとしても、その被害はパブリッククラウド事業者がすべて弁済してくれるためである。その場合、パブリッククラウド事業者としては、そのような状態になることを避けるため、必死で、自社の基盤ソフトウェアの脆弱性を発見・解消しようとするであろう。しかし、前述のとおり、ソフトウェアの脆弱性の発見・解消手段については、そのソ

フトウェアをバイナリであってもよいので公開し、多数の世の中の人々の手を借りて検証してもらう以外には、今のところ、少なくとも従来の OS 等の標準的なセキュリティレベルと同等の安全性を確保する効果的な方法が発見されていない。よって、パブリッククラウド事業者は、従来の OS 等のシステムソフトウェアと同等品質までセキュリティを高めるためには、いずれ、パブリッククラウド基盤のソフトウェア群を世の中に公開し、衆人環視によるセキュリティ検証を受ける必要が生じる。しかし、これは、先に解説したとおり、パブリッククラウド事業者の現在のビジネスモデル、すなわち基盤ソフトウェア群を非公開の状態に秘匿することにより、他のさまざまな会社が自社と同等のパブリッククラウドを構築することができる技術ノウハウを手に入れパブリッククラウド市場に参入することによる健全で公正な競争的市場の成立をできるだけ遅らせる努力を必要とするビジネスモデルの前提条件と矛盾する。これが現在のパブリッククラウド事業者のビジネスモデルの構造上の限界、セキュリティを高めて脆弱性を減らすという活動に対する限界である。パブリッククラウド事業者は、現在のビジネスモデルを変更してソフトウェアの公開主義への明るい進歩的転換を行なうか、あるいは、基盤ソフトウェアを秘匿したままにして重大な脆弱性を未発見・未解決の状態に留め置くかの、究極的ディレンマに陥っていて、この部分で進歩を停止してしまっている。したがって、当面は、パブリッククラウド事業者はソフトウェアを公開することができず、ソフトウェアに内在しているセキュリティ問題を従来の OS 等と同様の水準に高めるための改善は困難である。その間は、パブリッククラウド事業者は、SLA 契約において、損害賠償金額の免責を設定せざるを得ない。そして、SLA 契約において設定されている免責金額を除いた確実な賠償が約束されている金額のみが、パブリッククラウド事業者が実質的に自社の基盤ソフトウェアの安全性を自信をもって保証することができる度合いである。保証の程度がこのように限定的である以上は、ユーザーは、現在時点におけるパブリッククラウド事業者の基盤領域は、ゼロトラスト原則に基づく Untrusted Zone として扱い、Untrusted Zone 上に、さまざまなセキュリティ手法により Trusted Zone を作り出すという追加的手順を必要とするのである。

(4) クラウド・ショックの発生メカニズムとその対処方法の必要性

現代型の大規模集約型パブリッククラウドサービスには、現在の技術水準の限界により、以下の未解決のセキュリティリスクが存在する。

（ア）可用性リスク ―「クラウドサービスが壊れて、自社のデータが取り戻せない」

意外にも、データの所有権という法的概念は未だ存在しない。企業によるデータの権利を確実に保全するには、データを記録しているディスク装置という「物」の所有権を維持する必要がある。物の所有権は、「物権」と呼ばれる。物権は、大変強い。物が存在する限り存続する、絶対的な権利である。ディスク装置の物権としての所有権を確保しておくことが大切である。停電、災害、通信障害により、通常利用しているデータセンタが停止したとする。それでも、経営者は安心していられる。オンプレミス型システムでは、ディスクの所有権をユーザーが有しているからである。ユーザー企業は、堂々とデータセンタに入っていって (停電や認証システムが故障した場合なら、手動でドアを開けてくれるであろう)、自らのラックを開錠し、サーバーやディスクという「物」を取り出し、そのディスクを本社や庁舎に持ち帰ってデータを取り出したりして、業務継続が確実に可能である。極端には、データセンタ事業者がディスクを返してくれなければ、裁判所で返せという判決を得て、執行官を連れて行って、データセンタのドアをこじ開けてもらい、ディスクを取り返すこともできる。データセンタが倒産しても、ディスクは返ってくる。

これに対して、パブリッククラウドにおいては、ユーザーはデータをクラウドサービス事業者が所有権を有するディスク装置に書き込む。ユーザーはクラウドサービス事業者に対して、データを読み出しする権利を得る。これは物権ではなく、「債権」である。債権は、とても弱い権利である。履行してもらえるかどうか分からない。不履行があったら、裁判所に行って、履行せよという判決をもらうことはできる。しかし、それでも履行してくれない場合、現行法上、直接の強制をする手段がない。金銭による賠償か金銭制裁による間接強制で満足するしかない。クラウドサービス事業者が破たんしたら、賠償も得られず、泣き寝入りになる。

パブリッククラウドにおいては、サービス障害時にはデータの所有者は、データ

が全然取り出せなくなる。データを書き込んでいるディスクの所有権はクラウド事業者にあり、他のユーザーを含めた複数ユーザーのデータが、1 台のディスクに、高度に重畳的に多重化されて記録されている。データセンタの障害のほか、クラウド事業者の認証システムやメタデータ管理システムのコードに誤りがあったり、障害が発生すると、「クラウドサービスが壊れて、自社のデータが取り戻せない」状態が発生する。

（イ）機密性リスク ―「自社のデータが 10 年前から流出していた」

オンプレミス型システムでは、ユーザー企業が自らのコンピュータやディスクの所有権・支配権を有している限り、物理力 (例: データセンタへの設置と施錠) と自らの暗号化を用いて、自らの能力と責任で、確実に、機密性を維持できた。

パブリッククラウドにおいては、現在、同等の安全性は実現不能である。サーバーサイドで暗号化されて保管されているように見える全てのデータは、一見安全に見えても、クラウド事業者は、いつでも技術的に復号可能である。「毎回ユーザーが鍵を指定する方法で暗号化・復号化が行なわれる」という手法を用いても、データ暗号共通鍵そのものは一度クラウド事業者のサーバーのメモリに乗り、CPU で処理される以上、クラウド事業者は生データにアクセスできてしまう。復号化処理がクラウド事業者の有する CPU 上で行なわれる以上、クラウド事業者の最上位特権者 (基幹部分のプログラマ) は、データ暗号鍵 (これは共通鍵暗号で、ハードウェアセキュリティモジュール (HSM) を用いても保護不能である) にアクセス可能である。これを防ぐ技術的手段は、パブリッククラウドでは、実用化されていない。監査は、特権を有するプログラマに対して、ほとんど役に立たない。監査システムが監視できるのは、その監査システムの水準以下のオペレーションのみである。最上位特権者は監査システムをかわすことができる。これにより、クラウド事業者が暗号化されていないデータにアクセスことができる。また、認証部分の不具合により全世界にデータが公開されるリスクも存在する。クラウドサービス事業者の特権を有するプログラマが、過失または故意によりわずかな数ヵ所のコードの間違いを注入しただけで、これが発生し得る。「自社の預かる顧客データが、クラウ

ド事業者の故意又は過失により、10 年前から流出していた。」という事態が 10 年後にはじめて発見され、過去 10 年分の損害を顧客に賠償しなければならない事態が発生し得る。

上記 (ア),(イ) のリスクは、新規性のある話ではなく、クラウドサービス事業者や一定水準の技術者であれば誰でも知っている、既知の問題点である。しかし、ユーザー企業の経営者は知らない場合が多い。未だこの危険は現実化したことがないからである。これが現実化したときに多数のユーザー企業の経営者によって同時に発生するパニック的行動が、次の「クラウド・ショック」を引き起こす。

「クラウド・ショック」の発生リスクとそのメカニズム

大規模なパブリッククラウドにおいて、上記のリスクは、現在は幸運にも現実化していないが、2030 年・2040 年というような長い単位でみると、これらは現実に発生し得る。そのとき、きわめて危険な集団行動が発生する。上記のようなセキュリティ侵害が発生していることがついに表面化したとき、仮にその事件が小規模に発見された場合であっても、多くのユーザー企業は、できるだけそのパブリッククラウドから、いちはやくデータを避難させたいと考える。なぜならば、そのクラウド・スキャンダルはそのパブリッククラウド事業者に対して信用不安を引き起こすリスクが高いことは明らかで、信用不安が引き起こされたら、多数のユーザー企業がそのパブリッククラウドを利用しなくなるであろうことが各企業の経営者によって予測されるためである。それが発生すると、その大規模パブリッククラウド事業者の収入は激減し、事業継続の危機にさらされる。その大規模パブリッククラウド事業者は、データセンタの料金、電気代、回線費用、人件費を支払えなくなる。そうなると、サーバーが停止し、またメンテナンスが止まるので、データが取り出せなくなってしまう。支払困難になったパブリッククラウド事業者のサーバー群、ハードディスク群が、債権者に差し押えられるおそれもある。これは、最大限に深刻な事態である。前記のとおり、オンプレミス型システムであれば、データセンタが破たんしても、ユーザー企業は、所有権に基づいて自らのラックからハードディスクやサーバーを引き抜いて持ち帰ればよい (データセンタの物ではないので、債

権者は、差押えできない)。しかし、パブリッククラウドの場合、ディスク装置やサーバーはパブリッククラウド事業者の所有物である。決してユーザー企業の所有物ではない。信用低下によって支払困難となったパブリッククラウド事業者に対する債権者 (たとえば、従業員や、賠償請求権を有する顧客) が差し押えたら、債権者はこれを競売にかけて換金・売却し、お金を取戻そうとする。それ以外でお金を取戻す方法がないためである。ここまでで指摘したようなことは、ユーザー企業の経営者であれば、利用しているパブリッククラウドにおいて、いざ危機が発生したとき、誰でも瞬時に思い付くに至るであろう。もしパブリッククラウド事業者が事業停止する前までにデータを取り出しできなければ、ユーザー企業自らが倒産の危機に陥る。そこで、パブリッククラウドを信用して重要な業務データ・機密データを預けてきたすべてのユーザー企業たちが、いっせいにデータを避難させようとする。他のパブリッククラウドにデータを移行したり、もうパブリッククラウドは使いたくないと考えて、オンプレミスシステムを急いで構築したりしようとする。そのためには、これまで何十年間も長い期間をかけてパブリッククラウドに少しずつ蓄積してきたデータを、すべて短時間でダウンロードしなければならない。そして、それは他のユーザー企業よりも早くダウンロードしなければならない。他のユーザー企業たちのダウンロードが終わり、契約を終了したら、パブリッククラウド事業者の収入が途絶え、そうするとダウンロード不能になる。このようなパニック状態では、一刻も早くダウンロードをして脱出をしたほうが良いという具合になる。

　このとき、何が発生するだろうか。パブリッククラウドの構造は、集約多重効果を前提に構築されている。CPU、ディスク、ネットワークなどのハードウェアリソースと、ノード管理システム、メタデータ処理システム群などのソフトウェア機構とは、いずれも、多数のユーザーが同時にフル回転させることがない前提で構築されている。CPU 消費率に対して追加課金をしたり、ディスクのリクエストに対して課金をしたり、長期保管データの読み出しに対して高額課金をしたり、データ転送リクエストに対して高額課金をしたりするのは、各ユーザーにできるだけリソースの回転率を低下させてもらい、ハードウェア購入コストを最小化して、大きな利益を得るためである。できるだけぎりぎりのところでこれらの装置群 (固定資産)

回転させて利益を挙げ、また、パブリッククラウド間の価格競争による値下げを耐え抜いているのである。このような、決して多数のユーザー企業が同時にリソースやシステムに負荷をかけないことが想定されているシステムにおいて、上記のような信用不安が発生し、突然に多数のユーザー企業が同時にデータを待避し始めると、コンピュータ用語でいう、スラッシングに似た現象が発生する。CPU のコンテキストスイッチの切り替え、ディスクの I/O 待ち時間の増大、ネットワークの帯域不足、認証システムやメタデータ管理システムの高負荷による不具合の発生やメモリ不足が発生する。この中で特に最初に限界点を迎えると想定されるのが、ネットワークの帯域である。パブリッククラウド事業者の弱点は、ネットワーク帯域不足である。これは、インターネット接続系でも、プライベート接続ポートでも、変わらない。ネットワークはバースト的にデータが流れるので、高帯域化を行なうと、使われていない時間のコストが増大する。また、イーサネットの技術やルーティングの技術は発展途上であり、パブリッククラウド上の多数のユーザーの本来需要を同時に収容するだけのキャパシティのあるネットワークを、現在の技術水準では、十分に分散して処理することが困難である。この制限は、光伝送の速度、装置のコスト、装置の集約率、発熱の具合、ハードウェアレベルのロードバランス技術や方式の限界によって生じる。これが原因で、集約率が高く大規模なパブリッククラウド事業者であればあるほど、ネットワーク伝送量単価が高額に設定されているのである。これは、ネットワーク通信量で利益を得たいから高額に設定されているのではない。多数のユーザーが本当に大量のネットワークを利用したら技術的に全然耐えられないので、高額な単価を設定することにより、ネットワーク帯域の消費を節約するように工夫させているのである。

ある程度の数の企業ユーザーが一刻もデータをダウンロードしようとして、ネットワーク帯域、CPU、ディスク、メタデータシステムに負荷がかかると、上記のようなリソース不足により、そのユーザーのダウンロード速度は低下する。ネットワークボトルネックが発生すると、TCP のレイヤにおける再送が発生し、速度は急激に低下する。ディスクのボトルネックが発生すると、通常のディスクアクセスのソフトウェアの動作が遅くなったり、極めて不安定になり無応答に近くなる。この

現象は、パニックを呼ぶ。さらに多くの企業ユーザーの経営者たちは、データを一刻も早く待避させるように指示をする。もはやこの状態の経営者の心理としては、そのパブリッククラウドの欠陥によるセキュリティ侵害などは、どうでも良い状態であると考えられる。それよりもパブリッククラウド事業者が破たんするより一刻も前に、データを待避させなければならない。他の企業ユーザーたちが待避させて解約する前に、自社のデータを全部待避させなければならない。しかし、数十年かけて溜め込んできたデータストレージに置いたデータの待避には、相当な時間を要する。途中で、どんどんとデータ転送速度が低下していく。これはおかしい具合だぞと考えたユーザー企業の技術者たちは、その旨をインターネットの SNS 上で周知する。これがさらなるパニックを引き起こし、いよいよ、ほぼすべてのユーザー企業がデータ待避を決意する。そうなると、もうデータ転送速度は 80kbps くらいに低下し、全然ダウンロードできなくなる。ネットワークのパケットロスによる再送と、高度密集された CPU のユーザー間の時間の奪い合い、ディスク I/O 能力不足により、システム全体の負荷は常に 100% となり、データ転送は、ほとんど停止した状態になる。この状態になると、そのパブリッククラウドサービスの復活は困難である。データはもはや永久に取り出せないか、配給制のようになり、随分先に自分の番が回ってきて初めて取り出せる。ユーザー企業は、データ取り出しに成功するよりも先にクラウドサービス事業者のサーバー群、ディスク群を彼らの債権者が差押えられたり、クラウドサービス事業者の従業員が給与不払いで離散したり、破産手続に移行して競売にかけられたりしないことを祈るしかない状態に陥る。長期間の停止により、社会は麻痺状態に陥る。

「クラウド・ショック」への対処方法

われわれの社会は、いよいよパブリッククラウドシステムに依存しつつある。パブリッククラウドシステムは、クラウド・ショックの発生に対して大変脆弱である。したがって、われわれは、これを解決する技術を生み出す必要がある。クラウド・ショックのリスクを根本的に解決するには、誰でもパブリッククラウド事業者となり、パブリッククラウドを構築・運用することができる技術の発育と普及が必要で

ある。しかし、これには時間を要する。

そこで、現在は、クラウド・ショックがいつでも発生する危険性がある以上、これを緩和する現実的な対策手法が重要となる。具体的な対策手法としては、以下のようなものがある。

(1) 機密性を要する情報は、必ずクライアント側暗号化を施してクラウドストレージにアップロードする。このような暗号化を施していれば、クラウド基盤に対する脆弱性リスクが表面化した場合でも、自らは、慌ててデータを引き上げる必要がなくなる。攻撃者は生のデータを手に入れることができないためである。クラウド・ショックは、クラウドサービス事業者に関する技術上または経営上の信用不安が生じたときに、多くのユーザーが同時にデータを引き上げようとすることによって発生する。なお、仮にすべてのユーザーのクラウド上のデータがクライアント側での暗号化によって機密性が確保されているならば、仮にクラウド基盤に対する攻撃が成功しても、ユーザーは機密性に対して懸念する必要がなくなり、データを引き上げようとは考えなくなるので、クラウド・ショックの発生を、そもそも防止できる。

(2) 可用性を要する情報やシステムは、必ず手元にも、常時、ほとんど最新のデータのバックアップを用意しておき、パブリッククラウドが長時間停止した場合、直ちに手元の環境でシステムを構築・運用できるようにしておく。そのために必要なサーバーやネットワーク等の最小限度の構成を用意しておく。パブリッククラウド上で動作させている IaaS サーバーは、ディスクイメージを、必ず定期的に手元にバックアップしておく。クラウド・ショックが発生しても、自らは、慌ててデータを引き上げる必要がなくなる。手元のローカルバックアップを元に、手元のサーバーでシステムを再開することができるためである。クラウド・ショックは、クラウドサービス事業者に関する技術上または経営上の信用不安が生じたときに、多くのユーザーが同時にデータを引き上げようとすることによって発生する。なお、仮にすべてのユーザーがこのような対策を行なっているならば、ユーザーは可用性に対して懸念する必要がなくなり、データを引き上げようとは考えなくなるので、クラウド・ショックの発生を、そもそも防止できる。

すなわち、パブリッククラウドを利用する際には、その機密性・可用性を盲信することなく、常にデータのローカルバックアップをとり、同等のシステムを手元の環境を用いていつでも構築できる環境にしておくことが重要である。

ローカルバックアップが理想的であるが、場合によっては、他のクラウドサービスへのバックアップでも、一応のリスク軽減は可能である。同一のパブリッククラウドサービス事業者の異なるアベイビリティゾーンやリージョンにバックアップしただけでは、クラウド・ショックに対応することができない。必ずローカル、または他のパブリッククラウド事業者のストレージにバックアップする必要がある。

(5) クラウド上で構築または利用している各種サービスのクラウド・ショック発生時における運用継続性の確保

IaaS、PaaS、SaaS のいずれも、特定のパブリッククラウド事業者からデータを急いで避難されたのちに (それが困難な場合は、バックアップデータに基づき)、別のパブリッククラウド事業者のシステムまたはオンプレミスで、同等のシステムを、短時間で再現できるように、すべての手順書メモを残しておく。これが不可能な特別のパブリッククラウド事業者に依存するような特殊な機能は、決して利用してはならない (実験目的、研究開発目的はのぞく)。パブリッククラウド事業者のシステムで障害が置き、長期的に復旧できなくなったときでも安心で慌てることがない状態にしておく。

(6) ゼロトラスト原則の本質の理解と実現、表見的ゼロトラスト製品の敬遠

ゼロトラスト (Zero Trust) セキュリティモデルについては、正確な理解が必要不可欠である。ゼロトラストに関する形式的な理解、ゼロトラスト風な製品やクラウドサービスを導入すればゼロトラストが実現できるという考え方が散見される。ゼロトラスト風な製品を導入することにより、実質的に何らゼロトラストを実現しておらず、むしろ、かえって導入前の状態よりもセキュリティレベルが低下した状態となってしまっていることが多い。このような表見的ゼロトラストともいえる、ゼロトラスト製品やクラウドサービス群は、今は流行しているかも知れないが、そ

のうちその欠陥が明らかとなり、衰えてゆく。われわれは、そういった誤ったゼロトラスト風なシステムを構築してはならない。真のゼロトラストを実現しなければならない。

ゼロトラストの定義（1）— 米国 National Security Telecommunications Advisory Committee

Zero Trust is a cybersecurity strategy premised on the idea that no user or asset is to be implicitly trusted. It assumes that a breach has already occurred or will occur, and therefore, a user should not be granted access to sensitive information by a single verification done at the enterprise perimeter. ①

和訳: **ゼロトラストとは、いかなるユーザーや資産も暗黙のうちに信頼してはならないという考えを前提としたサイバーセキュリティ戦略である。**この戦略では、侵害がすでに発生しているか、または今後発生する可能性があることを前提としている。したがって、企業の境界で行われる 1 回の検証によって、ユーザーに機密情報へのアクセスを許可すべきではない。

ゼロトラストの定義（2）— 米国 National Institute of Standards and Technology

Zero trust is a cybersecurity paradigm focused on resource protection and the premise that trust is never granted implicitly but must be continually evaluated. ②

和訳: ゼロ・トラストとは、リソースの保護に焦点を当てたサイバーセキュリティのパラダイムであり、**信頼は決して暗黙のうちに与えられるものではなく、継続**

① The President's National Security Telecommunications Advisory Committee. Report to the President on Zero Trust and Identity Management. February 2022. https://www.cisa.gov/sites/default/files/publications/NSTAC%20Report%20to%20the%20President%20on%20Zero%20Trust%20and%20Trusted%20Identity%20Management.pdf

② NIST Special Publication 800-207 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

的に評価されなければならないという前提である。

ゼロトラストの本質

ゼロトラストの本質は、「**暗黙的な Trusted Zone は "可能な限り" 最小化しなければならない**」(the implicit trust zone must be as small as possible) という原則で表現される。これまでの社内 LAN 上の IT システムでは、LAN を Trusted Zone と考えてしまい、LAN 上のサーバーのセキュリティ対策を十分にとらなかった。ほとんど無認証でアクセスできているケースが多かった。これにより、Trusted Zone が広すぎて、攻撃者がつけ込む隙が多すぎるという問題が発生した。そこで、多くの人々がアクセスできてしまう LAN を Untrusted Zone と考えた上で、Trusted Zone を極小化し、Untrusted Zone と Trusted Zone との間に挟まるサーバーやサービスにおいて、必ずユーザー認証を施すべきである、というゼロトラストの考え方が提唱されたのである。

このゼロトラストというのは、何ら新しい思想ではない。画期的なものでもない。これはコンピュータ・ネットワークやシステム運用においては、本来当たり前の思想である。すなわち、ネットワーク上のサーバーを認証無しにアクセス可能な脆弱な状態に置かず、全サーバーにセキュリティをしっかりかけるべきというセキュリティ思想である。サーバーに接続し、データにアクセスする際において、必ずユーザー認証を施し、アクセス権限に基づいて許可されているデータのみにアクセスさせ、認証やデータへのアクセスのログをきちんと取るべきであるという考え方である。各サーバーで個別のローカル認証を行なっても良いし、LAN 上の共有ディレクトリ型認証システムを用いて複数のサーバーでシングルサインインを実装してもよいのである。そして、各サーバーで動作させるサービスは最小限とし、自動アップデートを施して、一応のぜい弱性対策をしておく。これでほとんど安全な状態となる。昔から十分なセキュリティリテラシがあるシステム管理者が、サーバー類のセキュリティ設定、認証設定をきちんと管理しておりさえすれば、たいていは、すでにゼロトラストが実現されている状態なのである。改めて梵鐘(ぼんしょう)を付いて騒ぐほどの問題ではない。ところが、この 20 年間で、ある程度の割合の IT システム

管理者が、最小限度のセキュリティリテラシの勉学すら怠り、「LAN にさえアクセスできれば実質的に認証なしで機密情報に内容に認証無くアクセスできてしまう」というような具合のシステムが、社内 LAN に乱立してしまったのである。にわかシステム管理者たちのやっつけ仕事の結果、そうなってしまっている。それでも、ファイアウォールや認証付きのリモートアクセス (VPN 等) を整備することによって境界を設け、LAN への外部からのアクセスを制限していれば、当初は安全であった。この場合、LAN 全体が Trusted Zone であるということになる。ところが、熱心な攻撃者やマルウェアが最近この境界を勢いよく乗り越えてくる場合がある。たとえば、LAN 上の別のシステムの端末にマルウェアが感染し、これがインターネット上の C&C サーバーと通信して、リモートコントロールできる状態になっているというような具合である。そうすると攻撃者は Trusted Zone に入ってくることができ、「横方向の移動」(lateral movement) により、LAN 上のさまざまな認証無しで侵入できる脆弱なサーバーを見放題となる。前記のような脆弱なサーバーが動いていれば、攻撃者が LAN にアクセスできさえすれば、そのサーバーのデータを読み書きできてしまう。

すなわち、一部の管理者たちは、物理的なネットワーク (LAN) 全体を Trusted Zone であると考えてしまったので、LAN 上のサーバーは脆弱でも良いという甘えの考え方が自然に生じてしまい、脆弱なサーバー群が台頭してしまったのである。これでは、攻撃者によって、LAN へのアクセスの認証が突破・回避されたら (物理的アクセスとしては、攻撃者の建物への立入り。論理的アクセスとしては、マルウェア等を介した LAN 上の 1 台の端末の乗っ取り。)、もはや Trusted Zone に接続されているこれらの脆弱なサーバー上の情報セキュリティが侵害されてしまう。問題点は、単純である。LAN 全体を Trusted Zone であると考えてしまった点にある。LAN はとても広く、これを Trusted Zone として保護することはもともと無理な話である。末端 1 点でも攻撃者が入り込むと全体が侵害されるためである。それでは Trusted Zone を完全に無くしてしまうことはできるだろうか。実は、これは不可能である。たとえば、大容量のファイルサーバーと背後のストレージ装置が LAN で接続されているとする。そして、ストレージ装置の制

限により、ファイルサーバーとストレージ装置との間では十分な認証がないとする。ストレージ装置の IP アドレスさえ知っていれば、iSCSI というプロトコルでアクセスできると仮定する。この場合、ファイルサーバーとストレージ装置との間のネットワークケーブルは、Trusted Zone である。なぜならば、いくらファイルサーバーに認証を施したとしても、サーバーとストレージ装置との間の LAN を攻撃者が支配すれば、ファイルサーバーを介さずに、ストレージ装置のデータをすべて読み書きできてしまうリスクがある。このように、Trusted Zone を完全に無くすことはできない。しかし、ファイルサーバーとストレージ装置との間のネットワークが、企業内 LAN のような広域なネットワークに接続されておらず、極小化された専用スイッチで接続されていれば、そこに侵入されるリスクは最小限まで (物理的にサーバー室に攻撃者が入ってくる場合と同様のリスクまで) 極小化できる。または、たとえ広域な LAN を経由していても、その部分が暗号化され十分な認証が施されているとか、VLAN バックボーンによって広域 LAN 上の一般利用セグメントと分離され、かつ、その VLAN バックボーンの基幹部分のスイッチは LAN 上の攻撃者によって通常侵入され得ない水準であるというような場合は、同様に、そこに侵入されるリスクは最小限まで (物理的にサーバー室に攻撃者が入ってくる場合と同様のリスクまで) 極小化できる。こうすれば、すべての攻撃者はファイルサーバーを経由しなければストレージ上のデータにアクセス不能となる。これは、ゼロトラストアーキテクチャに適合している。Trusted Zone が可能な限り最小化する (shrinking implicit trust zones) することに成功しているためである。

このように、ゼロトラストの本質を理解すれば、通常の水準のシステム管理者の視点で考えて、ゼロトラストの実現は何ら難しいことではないということが分かるであろう。すなわち、ゼロトラスト風な製品やサービスを買ってきて導入しなくても、ゼロトラストは実現できる。社内 LAN を WAN と同じくらい危険ですでに攻撃者が潜んでいるかも知れないと警戒した上で、従来の社内 LAN 上の適切な認証、アクセス制御、ログ管理がなされていないサーバーやアプリケーション群を、もう少しましな状態にセキュリティ設定すれば、それにより、ゼロトラストシステムは完成する。そして、新たなアプリケーションを組むとき、新たなサーバーを立

てるときは、最初から、ゼロトラストの考え方に基づき、設計、プログラミング、インストール、運用を行なえば良い。社内 LAN や、クラウドシステム上の VPC が、インターネット (WAN) と同等に危険であるという十分に警戒的な思想で組み立てられたシステムを用いている限り、その社内 LAN やクラウド上の VPC へのアクセスに従来型のセキュリティ (物理的な立入りの制限、ファイアウォールや VPN 等による保護) を施すことは尚良いことである。あえて社内 LAN やクラウド上の VPC を無防備の状態に晒す必要はないし、社内 LAN や VPC を廃止することもない。すべてのサーバーをクラウド上のグローバル IP アドレスにあえて乗せる必要もない。十分にゼロトラスト的に構成された社内 LAN や VPC 上のサーバーに対して、その内部ネットワークへのアクセスが一応従来型のセキュリティ機構で保護されていれば、そのサーバーは二重の保護機構で保護されている。攻撃者は 2 箇所の難関を突破しなければならない。その結果、セキュリティは最大限に高まる。重要なことは、このモデルにおいて、外側のセキュリティ境界である社内 LAN や VPC の領域を Trusted Zone として考えないことである。つい油断してこれらのネットワーク領域を Trusted Zone であるとみなしてしまうと、社内 LAN や VPC 上のサーバーの保護を怠ってもよいという自然的油断心情が発生し、いつの間にか、また脆弱な状態に戻ってしまうであろう。このように、ゼロトラストの実現と維持は、① サーバーやアプリ等のシステム構築の立場からは、LAN や VPC を Untrusted Zone であるとみなしてサーバー等に最大限の保護を施し、② LAN や VPC の管理者の立場からは LAN や VPC への侵入をできるだけ食い止めるよう努め、この異なる ① + ② の立場の両面で可能な範囲の対策を施すということが重要である。そして ② の完全な保護というのは不可能であるから、どちらかというと ① に重点を置くべきである。しかし、① を実現さえすれば ② の保護は消滅してしまっても良いというのは、単に、本来実現し得る高いセキュリティレベルを低く引き下げる手抜き的行為である。繰り返しになるが、重要なのは、① と ② をそれぞれ異なる立場でとらえて、両方ともセキュリティ向上を図ることである。① が ② に依存したり、② が ① に依存したりして、互いに相手方が何とかやってくれているだろうから安全だと誤信してはなら

ないという考え方が、ゼロトラストアーキテクチャの本質である。

ゼロトラスト原則は不朽であり、世の中の重要な統治的部分にはもともと備わっている考え方である。コンピュータシステムのセキュリティに関してみても、これは、極めて昔から存在した考え方である。IT におけるゼロトラスト原則の価値と重要性は、IT が存続する限り永続する。

だが、ゼロトラスト製品は、そうではない。これらの多くは、永続しない。そのうち動作しなくなる。それだけであれば良いが、むしろゼロトラスト原則の目的であるセキュリティ向上と逆の結果を生じさせる。十分に注意しなければならない。

「ゼロトラスト」と「ゼロトラスト風製品」とは異なる

「ゼロトラスト」と「ゼロトラスト製品」とは、明確に分離されて議論されなければならない。「ゼロトラスト製品」、特に「ゼロトラストを実現するための、パブリッククラウド的な製品」という製品群は、短期的なソリューションとしては一応表面的に動く場合があるが、そのうち、破たんまたは継続できなくなる、あまり頼りにならないのである。

むしろ、最近散見されるゼロトラスト製品を用いると、上記のゼロトラスト原則と逆行する結果が生じる場合が多いのである。すなわち Trusted Zone 領域が外部に拡大されてしまい、脆弱点が増え、攻撃者による情報漏えいや書き換えのリスクが極めて増大する結果となるケースが多い。たとえ OS ベンダがプログラミングしてこしらえたゼロトラスト製品であっても、同様である。ゼロトラスト製品は、IT、ソフトウェア技術、システムソフトウェア、セキュリティ等の理解が中途半端な者に対して、売り込みをかけ、それによってゼロトラスト原則に基づくセキュリティ強度が従来よりも高まると誤って認識させ、これを採用させる。これにより、ゼロトラスト製品の販売者は、2 つの利益を得ることができる。第一に、自社の製品にユーザーの IT システムを依存させることができ、売上が増える。しかし、この点は、他の製品 (たとえば、複合機とトナーの関係等) と変わらないから、新しいものではない。第二に、これが重要な点であるが、ゼロトラスト製品は、その性質上、ゼロトラスト原則的なアーキテクチャの中で、「Trusted Zone」という特権

領域に介入し、自らが特権領域における主人であるとシステム上振る舞うことが可能である。ユーザーの日常業務において取り扱われる情報や、重要な意思決定すべてに渡って関係するユーザーの生のデータが、当該ゼロトラスト製品の提供する「Trusted Zone」を通過する。そのようなゼロトラスト製品は、たいてい、ブラックボックス的性質を有する。まずここで初歩的な矛盾が露出する。Trusted Zone を極小化するためには、Trusted Zone の安全性は極めて高くなければならない。だが、ブラックボックス的性質を有する製品は、安全性の検証が困難である。この矛盾により、まず、ブラックボックス的ゼロトラスト製品から、ゼロトラスト原則と相反した結果が生じるという、珍妙な結果が生じる。だが、ゼロトラスト製品のリスクは、このプロプライエタリ性に留まらない。ゼロトラスト製品のブラックボックス的性質は、極めて強い外部的権力となり得るのである。人的組織であっても、IT システムであっても、いかなる統治権力機構のセキュリティであっても、その特権部分は最小化しなければならない (ゼロトラスト原則)。したがって、その特権部分に関与できる機会は、希少な財産的価値を有する。外来者は、通常は組織の中枢特権部分への関与も、接触も、許されない。これにより、組織のセキュリティは維持される。ところが、「ゼロトラスト」というラベルを貼り付けたブラックボックス構造のある外来的機構は、いとも簡単に、高セキュリティ組織の中枢部分に入り込んでくる。その外来的機構は、当初は外部業者的に振る舞う。すなわち、ユーザーの命令の通りに動くのである。しかし、時間が経ち、ユーザーがその外部的業者としてのゼロトラスト的製品に慣れてきたら、徐々に、その組織内における意思決定に関与し始める。主体であるユーザーは、これになかなか気が付かない。この特権への関与が可能な地位は、希少な財産である。この希少な財産として外部のブラックボックスが入り込むことについて、ユーザーの視点では、2 つの脅威がある。第一は、取引の対象となる点である。それはそれなりに脅威であるが、他の分野でも見られる現象である。第二は、その中央権力的ブラックボックスは、たいてい脆弱性の欠陥が存在するという点である。表見的ゼロトラスト製品の売り込みの動機は、表見的ゼロトラストを標榜することでユーザーのシステムの特権機構に近付き、その中心的存在に組み込んでもらうように画策し、その後で、その地位を利

用してさまざまなビジネスを行なうという商業的動機にある。したがって、他の製品よりもいち早く自らの製品がユーザーの特権領域に入り込むよう努力をしなければならない。迅速さがとても重要である。そうすると、どうしても、開発期間に無理が出てくる。ユーザー企業への介入競争に勝つためには、表見的ゼロトラストセキュリティを実現できれば、実質的セキュリティを実現できなくても良いので、表見部分を重視し、実質部分は後回しにされる。もともと表見部分は開発が簡単であり、実質部分のセキュリティ的部分の開発はコストがかかるから、経済原理からいって、どうしても、そのような結果となる。プロプライエタリ製品であれば、実質部分は十分に衆人環視されないので、開発者は、脆弱性が発覚するおそれがないと考えて、随分安心していてもよく、モラルハザードが発生する。これにより、ゼロトラスト製品や、それと対応するクラウド側ゼロトラスト的サービスそれ自体の中枢部分の脆弱性は検出されずに放置される。その脆弱性に気付くのは、高い技術を有する熱心なサイバー攻撃者のみである。これにより、サイバー攻撃者は、安全であると誤信されてユーザー組織の特権部分を司っている外来業者的な表見ゼロトラスト製品の特権的 Trusted Zone を掌握してしまう。表見ゼロトラスト製品をわざわざ入れなかった従来システムではもともと安全だったのに、わざわざそのような製品を招き入れてしまった結果、むしろ危険が増大してしまう。ゼロトラスト製品を作るような外国企業は、短期的利益、キャピタル・ゲインを求める投資家を満足させる必要があるので、長い目で見た安全で耐久性のあるシステムソフトウェアのプログラミングに割く時間に欠けるから、これは、やむを得ない現象である。そして、多数のユーザーによって使用されている OS 等と比較して、そのプログラムのバイナリコードは衆人環視されていない。被害に遭わないためには、ゼロトラスト製品なるものをできるだけ信用せず、そのプログラム内部に渡っての十分な検証をユーザー自らの手で行なう必要がある。

　このように、ゼロトラスト製品を入れれば、ゼロトラストが実現できるという訳ではなく、実際には全く逆であることが多い。ゼロトラストの実現においては、ほとんどの部分では、クラウドサービスやゼロトラスト製品の採用といった安易で刹那的な方法でなく、従来のシステムのセキュリティがあやうい部分を、ゼロトラス

ト原則に基づいて補強していく健全で建設的手法をとることが重要である。

もちろん、ゼロトラスト製品すべてが、上記のように脆弱なものであるという訳ではない。ごく一部のゼロトラスト製品は、真に実質的セキュリティを実現することを第一目標に、システムソフトウェアやセキュリティに関する長年の経験を有する少数の精鋭的プログラマが、一生懸命実装している。こういうものは、とても長持ちする。プログラムの重要性が大きい割に、脆弱性が少なく、安全である。ということは、ゼロトラスト製品を採用する場合において、大半の表見セキュリティ製品と、実質的に高い水準のプログラマが実装する長持ちセキュリティ製品とに峻別する必要がある。そのためには、ユーザー側において、かなり製品のコア部分のプログラムの内奥に踏み込んだ検証が必要となる。ユーザー側の見る目がなければ、製品のセキュリティ欠陥箇所の有無を評価することはできない。そのような検証・評価においては、単に他人が作ったシステムや製品を組み合わせてそれなりの見た目は機能するソリューションを作ることができる程度の中途半端な IT 技術者では不十分である。セキュリティを標榜していながら、自ら極めて多数のユーザーに利用され攻撃にさらされてもなお安全性を実現しているような重要でミッション・クリティカルなセキュリティ・ソフトウェアを、自らの手で開発したことがあるような経験が 1 回もないような、外観上のセキュリティ能力しか有さない素人的セキュリティ技術者では、不十分である。そのような人材は、民間にありふれている。大企業や政府部門の経営者たちは、これまで技術者を見る目がなかったから、そういった、自ら手を動かしてコードを書き攻撃者による大量の攻撃に自らのコードを晒したことが一度もない表見的セキュリティ技術者を民間から雇って、そういう人たちに製品選定をさせたりするのである。そういった技術者は、大企業の管理職や、政府の官僚を煙に巻き、自らの能力の不十分な点はうまく取り繕い、自らの気に入ったベンダや製品を偏向的に推奨する傾向にある。結局はそういった製品がむしろセキュリティ事故を引き起こすのである。そういった事故は、企業や政府におけるいわば自損事故である。組織内の問題である。だが、扱う情報が国・地方自治体における安定的な統治に直結する場面においては、影響は国民に及ぶから、こういったことは決して発生しないように、予防策を考える必要がある。これを防ぐために

は、どのようにすれば良いか。前記のような、システムソフトウェアプログラミングと呼ばれる領域に精通しており、自らの手で、数十万人、数百万人の重要なセキュリティ領域において長年利用され、そのコードからは深刻な脆弱性が少なく、長期間安定して継続しているソフトウェアの開発者たちを集めることが必要である。ある対象製品と同質のソフトウェアを、自らの手でプログラミングし、配布し、大量の攻撃に耐え、鍛えられた経験を有するセキュリティ人材でなければ、その対象製品を正しく評価することは、不可能である。そして、多様性を確保するため、そういった人材はできるだけ多く集める。そういった人材からの衆人環視によるディフェンスに耐性があるゼロトラスト製品のみが、安全な製品である。そのようなゼロトラスト製品のみを、政府や地方自治体は、はじめて、安心して導入することができる。そのような多様性に基づく安全検証をクリアしなければ、ゼロトラスト製品を導入することによる危険性は、導入しない場合と比較して利益を上回る。繰り返しであるが、ゼロトラスト原則は、思想・アーキテクチャであり、製品のことではない。自らの手で安全性を検証したゼロトラスト製品が充実してくるまでの間は、機密性が高い情報を取扱う領域においては、ゼロトラスト製品を早計に導入するべきでない。失うものが大きすぎるためである。従来型システムにおいて認証・認可・ログ記録を怠っていた部分にゼロトラスト概念を導入して安全化する手法によるべきである。

(7) パブリッククラウドの利用のゼロトラスト原則に対する本質的危険性と契約上の緩和策

上記のようなゼロトラスト原則に導かれる必要最小限の対策は、クラウド自由に対する制約を伴う。これに対して、パブリッククラウド事業者が、ある局面（彼らの内部の Trusted Zone）について十分なセキュリティの保証を提供できると主張するならば、以下のような条件を満たす限り、例外は許容され得る。

(ア) そもそも、利用契約上に、損害賠償の金額や期間の上限や免責（特別損害の除外など）が定められている限り、彼らはその賠償範囲に相当するわずかな部分だけ責任を負っていることになる。パブリッククラウド事業者は、安全性について、内心かなりの不安があるから、そのような免責規定を掲

げているのである。

たとえいかなる高水準のプログラマが集結しているパブリッククラウド事業者であっても、現在の技術水準的では、十分なセキュリティの保証は、相当困難である。そして、顧客からはそれを確認できない。したがって、本当に確実に保証を提供できるとするのであれば、担保が必要である。クラウドサービス事業者のセキュリティ能力は十分保証できると言っている個人や法人がいれば、その個人や法人が保証人になるべきである。その保証人の信用資力と、債務不履行があった場合のそのパブリッククラウド事業者に対する財産の強制執行・保全の可能性が十分であり、かつ、これが客観的な資料で明らかである場合に限ってはじめて、そのパブリッククラウド事業者の Trusted Zone は信用できるということができる。

(イ) 現状、いかなるパブリッククラウド事業者であっても、自社の基盤システムについてその Trusted Zone に関して十分に安全・安心であるといえる技術水準には達していない。規格を表見的に満たし、監査を表見的に通過していることは、「安全水準に達している」ことの担保にならない。そこで、本来は、ユーザー側は、その中身を自らの目でみて、検証することが必要である。そのために、パブリッククラウド事業者によるパブリッククラウド基盤のソースコードのオープンソースは、理想的であるが、現実には、この部分を秘匿することにより排他的利益を得たいというのが彼らのビジネスモデルであるため、現実には、ソースコードの客観的安全検証は、難しい。ところで、ソースコードの公開は、セキュリティの検証において、実のところ本質的必須要素ではない。従来の、ソフトウェアのバイナリコードがユーザーの手元で動作する方式のソフトウェアについて考えると、プロプライエタリ製品を用いる場合であっても、検証は十分に可能であった。従来型の OS やオンプレミス型のシステムソフトウェアにおいては、ソフトウェア資材がバイナリしかなくても、少なくとも同一のバイナリは多数のユーザーの手元にあり、また無数の研究者が同時並行して分析しているので、かなりの水準の欠陥の指摘が可能であった。ところが、パブリ

ッククラウドのソフトウェアはバイナリすら隠蔽されているので、これが不可能となってしまった。ユーザーの衆人環視による安全性の検証作業(欠陥の不存在の確認作業) が、バイナリを通じてでも可能であるか、それとも、全く不可能の完全ブラックボックスクであるか、という点が、従来型 OS 等のソフトウェアと、パブリッククラウド制御基盤などのクラウドソフトウェアとの、重大な相違点である。パブリッククラウド事業者においては、現状、「決して外部に発覚することがないリコール事故隠し」が可能となってしまっている。このような衆人環視下にない状態では場合、技術者や経営者たちは、"必ず"、モラルハザードを起こす。指摘されるおそれがない場合、リスクの高い状態に放っておくのである。このようなソフトウェア欠陥の無責任な放置の可能性に対して、「それは、監査によって防ぐことができるから、安全である。」という盲信的主張が存在する。しかし、パブリッククラウド事業者の基幹部分の特権を有するソフトウェアのプログラマたちが次々と実装し、変更し、運用するコード群に対して、セキュリティ監査は、ほとんど役に立たない。監査を受ける側のパブリッククラウド事業者の基幹部分の特権を有するソフトウェアのプログラマたちは、その技術水準は世界最高級である。それでもセキュリティ上の問題をひんぱんに埋め込んでしまう。それほどセキュリティ対策は難しいのである。ところで、彼らのソフトウェアに関する監査は、同等の「世界最高級の」水準を有する多数の監査員によって監査される必要がある。普通の人材市場に存在する上級適度の技術者には、彼らのプログラムコードは十分に理解できないためである。ところが、パブリッククラウド事業者自らが外注する閉鎖的な監査会社が雇用する人材には、パブリッククラウド事業者の基盤部分のプログラムコードを監査することができる水準の人材は存在しないか、仮に幸運に存在していても、人数が極めて少ない。世界最高級の水準を有する技術者たちが書く高度複雑なソフトウェアの監査は、全く同等程度の外部の技術者たちの衆人環視によってのみ行なわれる。伝統的な大規模プロプライエタリな OS (たとえば、バイナリしか提供され

ない Windows、macOS、Cisco の IOS 等) においても、Microsoft 社員、Apple 社員、Cisco 社員たちと全く同等程度の外部の多数の技術者たちの衆人環視が行なわれている。これにより Microsoft 社の Windows、Apple 社の macOS、Cisco 社の IOS などは、バイナリのままでも、そのバイナリを衆人環視によって分析することによって得られたセキュリティ問題が指摘され、修正され、保護されてきた。この好ましい流れが、パブリッククラウドの基盤ソフトウェアでは、発生し得ない。バイナリすら公開されることがなく、独立的立場による高度技術者によるセキュリティ衆人環視的監査の実施が不可能となってしまっている。完全なブラックボックスが存在し、その瑕疵を発見して指摘する者がいない。これにより、パブリッククラウド事業者の基盤システムには、従来の OS やネットワークやセキュリティ基盤システムと比較にならないほどの脆弱性が存在し、これらが指摘されることなく放置された状態が形成されてしまっている。この重大なパブリッククラウドのセキュリティ問題に対する解決手法は、今日時点では、存在しない。

(ウ) このように、とても重要な共通的セキュリティ基盤の根本部分が脆弱であるとき、能力の高い攻撃者は、ここに目を付けて、一瞬ですべてを掌握できる。能力の高い攻撃者とは、パブリッククラウド事業者の内外に精通していて内部事情に詳しい攻撃者であり、国家レベルの攻撃者である。西洋主義国 (欧米や日本) に対して、緊張・対立関係にあるような、さまざまな国々の攻撃者である。そのような高度な攻撃者は、すでに発見している脆弱性を用いて、10 年くらいの時間をかけて、突然に大規模なパブリッククラウド基盤に対して攻撃をする。このような、パブリッククラウド事業者の基盤システムがサイバー攻撃を受けるリスクは、多くの賢明な企業経営者によっては、既知の問題であり、十分顕名な経営者たちには、すでに明確なリスクとして、認識されている。したがって、多くの名高い民間企業は、パブリッククラウドには「あまり重要でないが多数のユーザーが同時にアクセスする公開系のシステム」(たとえば、公開 Web サイト) を

置くことはあっても、停止してはならないシステム、機密性を維持しなければならないシステムといった類の重要システム群は、決してパブリッククラウド事業者に頼らずに、従来の SI 的に社内でプライベートクラウドを構築したり、従来型のデータセンタサーバーで運用したりすることにより、正しくリスク分離を図っている。

(エ) このように、民間企業の経営者においても、一極集中パブリッククラウド基盤のリスクを認識して、正しくリスク分散を行なっている。パブリッククラウド事業者を過信することなく、パブリッククラウド事業者の基盤システム領域はすべて Untrusted Zone であると想定した上で、重要なシステムは決してパブリッククラウド基盤に依存せずに安全に分散構築運用している。パブリッククラウドは、致命的でない点でのみ利用し、便利な特徴のみを、つまみ食い的に利用する。これが、パブリッククラウドの正しい活用方法である。

(オ) 日本国の行政という仕事は、すべての民間民間企業の活動すべてが乗っている統治基盤を維持するという、より基盤的・共通的な事業であるから、われわれは、各企業の前記のような堅実なセキュリティ・リスク管理よりもさらに一段と厳しくリスクを評価し、一般的企業以上に、パブリッククラウドの利用、パブリッククラウド事業者の Trusted Zone を真に信用することには、さらに慎重となり、その程良い健全精神状態を長年にわたって維持しなければならない。日本国の安定した統治を保障するために、できるだけすべての行政 IT 関係者が、セキュリティの本質、ゼロトラストの本質を理解することが必要である。しかし、それには時間がかかる。だが、少なくともわれわれのような類の役割を持っている者たちは、セキュリティの考え方、ゼロトラストの考え方を決して忘れてはならない。そして、その考え方を整理して体系化・文章化し、国と地方行政の将来を担う方々に対して時間をかけて案内してゆかねばならない。

(カ) パブリッククラウド事業者は、単なる有限責任の民間会社に過ぎない。民間会社は、経営者たちが完全に排他的支配管理をしている。民間会社は、

お金があるように見えても、いつでも配当ができるし、任意の外国に容易に資産を移転できる。いざ問題が生じたとき、他にも大勢の債権者が発生する。クラウド・ショックが同時的に発生して損害が莫大となったら、われわれユーザー (日本国・地方自治体) がパブリッククラウド事業者から得られるべき損害賠償請求権は、たとえ訴訟で勝っても、物理的にお金がないからという理由で結局払われない。したがって、パブリッククラウド事業者からの被害弁済は、ほとんどあてにすることができない。

(キ) ただ、パブリッククラウド事業者の債務履行に問題が生じた場合に備えて、会社取締役など連帯保証人を立ててもらえるならば、それは保証人たちの資力の範囲内で、一応安心できる (法律により、保証は書面で明示的になされなければならない。単に「私が保証します」というだけでは、口頭の保証契約なので、無効である。また、パブリッククラウド事業者の経営者等の個人が保証人となる場合は、極度額の明示が必要である。詳細は、後に述べる。このような連帯保証契約は、一般的な企業では、よくみられるである。仮に、パブリッククラウド事業者の Trusted Zone のセキュリティを全面的に信用して、そこが崩れないことの前提のシステムを立てるのであれば、少なくとも、そのような保証がなされることが最低限の条件となる。なぜなせば、パブリッククラウド事業者の Trusted Zone に欠陥があり (これは、ソフトウェアだから存在する)、そこに攻撃者が入り込んで横展開をし、同一のプラットフォームを利用しているすべてのユーザー組織に影響が生じたら、その被害額は甚大であるためである。そのような被害が発生し、われわれ (国・地方自治体) がその損害を被ることがないように、パブリッククラウド事業者には最大限の注意を払ってもらわなければならない。しかし、パブリッククラウド事業者の経営者たちが何らの担保も提供しないのであれば、経営者たちが、そのような注意を怠ることは、当然のことである。われわれユーザー側は、彼ら経営者たちが保証を提供する範囲でのみ、そのパブリッククラウドの Trusted Zone の安全性を信用してよい。仮に彼ら経営者が誰も保証を提供してくれないのであれば、

そのパブリッククラウド事業者の Trusted Zone の信頼性は、それなりのものである。なぜならば、そのパブリッククラウド事業者の Trusted Zone が万全なものであれば、経営者たちは保証を提供することに躊躇しないはずだが、それができないということは、Trusted Zone が万全なものではないことが、健全な企業感覚を有する者であれば、誰でも、推認できる。パブリッククラウド事業者において、規模が大きく、また、今のところ集まっている私的技術者集団の能力もそれなりに高いという評判がある場合、パブリッククラウド事業者に関してその信用を考えるとき、われわれは、健全な間隔が麻痺してしまう傾向がある。しかし、落ち着いて考えると、パブリッククラウド事業者の民間企業には、何ら特殊な点はない。超人が経営している訳でも、技術開発している訳でもない。彼らは人間である。過失もするし、故意で不正を行なうこともある。だが、法人は有限責任であり、経営者、従業員、株主は、法人と責任の点で切り離されている。法人が十分な賠償を提供できない限り、重大な過失のような任務懈怠責任があったことを、われわれユーザー側が証明しない限りは、経営者に対して賠償請求ができない。そして、パブリッククラウド事業者の内部にその立証のための証拠が残っていても、その証拠を取り出す手段がない。よって、ユーザー企業は、いざ事が発生したときも、パブリッククラウド事業者の経営者たちの責任追及が極めて困難である。加えて、重過失でないほとんどの過失に対しては、責任追及がもともとできない。これは、経営者集団にモラルハザードを生じさせる。不正の意図がない経営者集団であっても、過失は引き起こすので、長期的に必ず事故が発生する。過失による事故で、ユーザーであるわれわれが被害を受けることは、避けることができない。よって、われわれユーザー側は、その被害を極小化するため、取り扱う情報に求められる完全性・機密性・可用性が高い場合は、クラウド事業者が安全であると主張する彼らの Trusted Zone は Untrusted Zone であるとみなす必要がある。そして、ゼロトラストの考え方によって、暗号化や、VM (CPU) とデータの保管事業者の分離といった、"可能

な限り" の Trusted Zone の極小化を実現する必要がある。あるいは、前記のとおり、クラウド事業者の経営者たちに保証人になってもらい、担保を差し出してもらうことである。そうすれば、その担保の範囲内で、クラウド事業者の主張する Trusted Zone は、真に Trusted Zone であると信頼しても、安心である。担保があるためである。

(ク) 「パブリッククラウド事業者 X 社のクラウド基盤が提供する Trusted Zone 領域は、安全です。信用できます。私がその信用を保証します。」という個人がいれば、その個人には、このような会議に是非とも出席いただくべきである。そして、その場で、その個人に、国と各地方自治体に対してこれを保証をする旨の書面を書いていただくことである。口頭だけでは、保証は成立しない (民法 446 条 2 項) からである。また、損害賠償も含めて包括的に保証してもらう必要があるが、どの限度で保証するのか、金額としての極度額も明記しなければならない。極度額を明記しなければ、包括的な保証は無効である (民法 465 条の 2 第 2 項)。このような書面なくして、「このパブリッククラウドは信用できる」、「私がこれを保証する」と述べるのは、担保がなく、無意味である。セキュリティ (security) という英語の主要な意味は、「担保」である。誰も担保しようとしない限り、パブリッククラウド事業者のクラウド基盤が提供する Trusted Zone 領域は、ユーザーからみると、Untrusted Zone としてみなす必要があり、クライアント側暗号化やマルチクラウド等の手法を併用しなければ、重要・機密の情報の保管や処理には、利用できない。

(ケ) 「確かに、パブリッククラウド事業者は契約上の義務または損害賠償義務を履行してくれないかも知れない。しかし、米国系のパブリッククラウド事業者は、例外的に安心である。」という、非合理的で、米国人ですら有していないような謎の盲信を行なう日本人の技術者陣が存在する。しかし、米国の民間パブリッククラウド事業者が、米国系であるという理由で、安心できるという理論は、存在しない。米国政府がパブリッククラウド事業を行なっているのであれば、日米政府間同盟関係に基づき安心かも知れな

いが、「米国系パブリッククラウド事業者」とは、米国政府とは何の関係もない。よって、米国だから安心だという理論は成立しない。

米国政府は日本の同盟国政府である。米国政府が日本を裏切ることはないであろう。そうであっても、米国系パブリッククラウド事業者が日本国と何ら同盟関係にあるとはいえない。「米国のパブリッククラウド事業者は特に安心であり、日本人に対して、裏切らずに、最後まで、約束を履行してくれるであろう。」と誤信することは、十分な思慮を巡らしていない技術者にとっては、やむをえないことである。もちろん、日本は長年、米国政府と同盟関係にあり、このことは、今後も長年、変わらないとしても、そのことと、単なる民間株式会社である「たまたま本社が米国に設立されたパブリッククラウド事業者」の信用資力 (= 日本政府がユーザーとなった場合に、日本政府を裏切らないという信用) とは、何ら関係がない。「米国の事業者だから約束を履行してくれるであろう、何かあっても米国政府が保証してくれるから安心安全であろう。」、という理論は、米国政府という法人と、米国系パブリッククラウド事業者という法人が、完全に別々の「他人」であり、かつ、前者の人は後者の人の債務を保証してくれる訳ではないという、重大な点を見落としている。日本政府は、米国政府を信頼する。しかし、その米国政府が、米国クラウド事業者の債務を裏書保証してくれる訳ではない。仮に日本政府と米国系パブリッククラウド事業者との契約関係において、何か事が起こっても、米国政府は、これは民間事業者のことだから知らないと言うであろうし、それは当然のことである。前記のとおり、米国政府と、米国系パブリッククラウド事業者とは、「他人」である。他人の保証を自ら引き受けない限り、責任を負うことはない。このように、クラウドサービス事業者は、単なる有限責任の民間業者である。日本企業よりも歴史が短く、人的つながりも薄い外国人である。表見的に、お金があるように見えても、すぐに配当したり、世界中に財産を分散されたりできる。そうすれば、もはや、責任追及はできない。

(コ) クラウド・ショック等の発生時には、このような危険が表面化する。これ

は、一極集中が進むと、いつでも発生し得る。2030 年・2040 年までに発生し得る可能性はかなり高い。短期的な物事を考える場合、このようなクラウド問題の影響は限定的である。しかし、本会議は、長期的方針を考える会議である。この議論は、明らかに、本会議のスコープ内である。

(サ) われわれは、なぜか、米国系パブリッククラウド事業者について考えるとき、この有限責任の制約を、ついつい、忘れてしまう。彼らは新興事業者である。スタートアップ・ベンチャーである。株主利益追求型の私的技術者集団である。われわれは、彼らのパブリッククラウド事業者基盤に、決して、依存しない。われわれの業務に必要な IT システム要素群を、全面的に、彼らのパブリッククラウド基盤の上に構築してしまい、彼らの営業上都合の良いようにロックインされる状態、彼らのパブリッククラウド基盤の Trusted Zone 部分にサイバー攻撃がなされ、それがすべて Untrusted Zone となる瞬間にわれわれの全システムにサイバー攻撃者がいつでも触れられる状態という脆弱な状態に、決して、さらすことはない。雲 (cloud) は、大地 (ground) の周辺部分に過ぎない。われわれは、システム国土の大半、コアの部分を、揺れ動くことがない大地 (ground) に構築し、これにより堅牢なシステム王国を形成する。クラウド (cloud) は、揺れ動いてもよい、不安定性が許容される、周辺部分の断片的なシステム要素として利用するのみである。

(シ) われわれは、彼らのパブリッククラウド事業者基盤を、われわれにとって有利な点でのみ、部分的に利用する。パブリッククラウド事業者のシステムをいわば「こき使い」、彼らのパブリッククラウドリソースから、「われわれにとっての最大の利益」を引き出す。とても安価な料金で、通常は滅多に侵入されたりダウンすることがないことが期待されるような、公開し続けることが重要なそれほど機密でない情報 (たとえば、Web サイト、DNS サーバー等) をパブリッククラウド事業者の基盤で動作させる。滅多にダウンしないというとても苦労する負担をパブリッククラウド事業者に担ってもらう。その代わりに、報酬を支払う。パブリッククラウド事

業者複数社でこの報酬金額を競争させ、できるだけ安価な金額を実現してもらう。われわれは、ユーザーとして、パブリッククラウド事業者間でいつでも乗り換えることができるように運用し、交渉をする。各パブリッククラウド事業者が、われわれに対して、できるだけ高い負荷と責任を、できるだけ安い金額で請け負ってもらえるように交渉をし、競争を加速させる。われわれは、これにより、大きな利益を得る。各パブリッククラウド事業者も、最適化が進み、さらなる技術開発が進む。健全な進化の流れが生じる。

2　各論

(1)　クラウドストレージやクラウド上のファイル共有サービスの利用時にはクライアント側暗号化が必須である

機密情報を、クライアント側暗号化を施すことなく、クラウドストレージやクラウド上のファイル共有サービスにアップロードする利用方法は、絶対に避けるべきである。クラウド側での暗号化は、表見的セキュリティであり、実際には、機密性は保護されていない。

クラウドサービスにおけるファイルストレージ (オブジェクトストレージ) や SaaS 的なファイル管理・共有アプリシステムは、ストレージ管理の煩わしさから解放されるので、便利である。ユーザーが手動でこういったシステムを利用してファイルを共有することも、プログラマが API を用いてオブジェクトストレージの読み書きをすることもある。いずれも、開発コスト、運用コストが大きく削減でき、素晴らしいものである。

しかしながら、ユーザーも、プログラマも、こういったクラウドストレージを利用する際には、必ず、重要な機密情報をクライアント側で暗号化した状態で、その暗号化されたデータをクラウド上にアップロードすることを条件付ける必要がある。クライアント側での暗号化というのが、とても重要な点である。クラウドサーバー側での暗号化ではない。クラウドサーバー側での暗号化は、機密性の確保上、あまり意味がない。ここで暗号化について整理する。クラウドストレージを利用する場合の暗号化手法は以下の 3 種類の方法がある。

(1) クライアント側での暗号化・復号化 (鍵はいかなる場合でもクラウド事業者に渡らない)

(2-a) クラウドサーバー側での暗号化 (クラウド事業者が自ら鍵を生成し、その鍵によりクラウド事業者の CPU 上で暗号化・復号化処理がなされる方式)

(2-b) クラウドサーバー側での暗号化 (ユーザーが鍵を生成し、アップロード・ダウンロードの都度、その鍵をクラウド事業者に提示し、クラウド事業者の CPU 上で暗号化・復号化処理がなされる方式)

ゼロトラストの原則である、「Trusted Zone は可能な限り最小化しなければならない」(the implicit trust zone must be as small as possible) という考え方からすると、(2-a) が機密性の保護につながらないことは明らかである。すなわち、2-a は、クラウド事業者のインフラストラクチャの特権部分全体が Trusted Zone であるということになってしまう。ゼロトラストの原則では、このようなとても広い Trusted Zone はすでにセキュリティ侵害されているとみなさなければならない。すなわち、クラウド事業者のインフラストラクチャの特権部分という広い Zone に対して、攻撃者が潜んでいる可能性を想定しなければならない。攻撃者は、(2-a) の鍵にアクセスできる。したがって、攻撃者は、いつでも、暗号化されたデータを読み出すことができる。これでは、機密性が保護されていることにはならない。ここまでは、誰にでもすぐ認識できるセキュリティ脅威である。そこで、ほとんどのユーザー側設計者たちは、(2-b) の暗号化を利用しようと考える。

クラウド事業者のストレージのセキュリティに関するドキュメントを軽く読んだだけでは、上記の (2-a) と対比して (2-b) はユーザーが鍵を毎回指定するので安全であるかのように見える。しかし、ゼロトラストの原則から、クラウド事業者のインフラストラクチャの特権部分全体というとても広い領域に攻撃者が潜んでいることを想定することになる。この場合、想定される攻撃者は、毎回のリクエストとレスポンスの際にユーザーが提供する共通鍵を複製することができる。また、CPU で暗号化・解読処理がなされるので、暗号化前または解読後のデータがユー

ザーと送受信されるタイミングで、これを直接読み取ることも可能となってしまう。すなわち、(2-a) と (2-b) との Trusted Zone の広さは、攻撃者の視点からみて、実質的に、大きく変わらないのである。

ゼロトラストの原則では、「Trusted Zone は可能な限り最小化しなければならない」というものであるから、われわれは、クラウドストレージを利用する場合の暗号化手法としては、必ず (1) の手法を (2-a) または (2-b) と組み合わせて利用しなければならない。(1) によりクライアント側で暗号化を行なえば、クラウド事業者のインフラストラクチャの特権部分という広い Zone は Untrusted Zone であるとみなすことが可能となる。なぜならば、クライアント側暗号鍵はいかなる時点でもクラウド事業者のインフラストラクチャを通過しないため、当該 Zone に潜んでいる攻撃者がいかように横展開的な活動しようとも、ユーザーの情報を無断で読み取り、または無断で改変することは不可能となるためである。この状態では、依然として、その攻撃者はデータの消去・破損のみを引き起こすことができる。しかし、これは可用性の問題であり、機密性に対する影響はない。機密性が維持されていれば、少々可用性が低くても、パブリッククラウドサービスの利用は、許容され得る。なぜならば、複数の事業者のパブリッククラウドに渡って、または、オンプレミスを活用して、バックアップを行なうことで、データ消去・破損に対応することができるためである。

ゼロトラストの原則は、「暗黙的 Trusted Zone は "可能な限り" 最小化しなければならない」と義務付けている。"可能な限り" とあるとおり、ゼロトラスト原則は、不可能を強いているわけではない。クライアント側での暗号化のコスト、オーバーヘッド、容易性について考えてみる。クライアント側での暗号化は、とても簡単で、処理速度に対する影響もほとんどない。AES-256 や ChaCha20-Poly1305 といった政府推奨暗号リスト[①] で暗号化すればよい。一般的なラップトップでも、数 Gbps の速度で、スマートフォンの CPU でも数 100Mbps の速度で、暗号化・復号化が可能である。ネットワークの速度は有線 LAN であってもせいぜい数

① https://www.cryptrec.go.jp/list.html

100Mbps であるので、暗号化・復号化がボトルネックになるということはない。暗号化によって生じるデータ量の増大は、ヘッダやパディングと呼ばれるわずかな量 (数十バイト程度) であり、データ保管コストが増えることもない。クライアント側暗号化を困難とする技術上の要因は見られない。クライアント側暗号化によって "可能な限り" Trusted Zone を最小化するというゼロトラストの原則をはじめて実現することができる。したがって、ゼロトラスト原則を用いる限り、クライアント側暗号化は必須である。

クライアント暗号化の実装は、関数 1 つを呼び出すことで実現できる。組織間のファイル共有アプリがクラウドストレージを利用する場合も、クライアント暗号化を実装することは容易である。クライアント側暗号化に対応していない、レガシーなファイル共有同期アプリ (バイナリのみ公開されており、クライアント側暗号化をユーザーが実装できない) を利用する場合も、フィルタドライバと呼ばれるプログラムを実装すれば、レガシーなファイル共有同期アプリに渡されるファイルデータを透過的に暗号化することができる。そのようなフィルタドライバを実装してある日本製、海外製の多様の安価で信頼できるクライアント暗号化フィルタドライバ製品も充実しており、日本においては、大企業を中心に普及している。

最も重要なこととして、クライアント側暗号化を施した場合でも、その暗号鍵は、決して、誤って同じクラウド上のストレージ上に置いてはならない。仮にその暗号鍵を誤って誤って同じクラウド上のストレージ上に置いてしまうと、それは、金庫の上に金庫の鍵が置いてあるのと同様となり、(2-a)、(2-b) と同様に、攻撃者がその鍵にアクセスできてしまい、データの復号化が可能となり、無意味である。クライアント側暗号化の鍵は、必ず、クライアント側の端末やサーバーにのみ保持されなければならない。

ところが、ここで 1 つ問題が発生する。「クライアント側暗号化」の暗号処理が、オンプレミスのコンピュータ、すなわちユーザー企業が排他的支配権を有するコンピュータ上で実施される場合は、その排他性を有する限り、ほぼ 100% の安全性が実現できる。他方で、当該処理がクラウドサービス事業者の別のサービス上の VM 基盤上で動作するときは、やはり攻撃者はその CPU が処理する瞬間の秘密

鍵または生データに容易にアクセスできてしまう可能性が存在する。このリスクをどのように考えるかが問題となる。

ある Web アプリケーションを IaaS または PaaS としてのクラウド上のプログラムとしてデプロイし動作させるとき、当該アプリケーションは、クラウドサーバーの VM の CPU を用いてクライアント側暗号化を行なうことになる。ゼロトラストの原則をあてはめると、すでに攻撃者はクラウド事業者のインフラストラクチャの特権部分という広い Zone に侵入していることを考慮しなければならない。単一のクラウド事業者の IaaS または PaaS の VM 基盤があり、同時に同じクラウド事業者のオブジェクトストレージ基盤がある場合、これらの 2 つの Zone に対して、攻撃者が同一の特権 (当該クラウドサービス事業者の十分な技術的権限がある特権的開発者と同様の権限) でアクセスできる状態があるかどうかが、IaaS または PaaS の VM とオブジェクトストレージとを当該単一のクラウドサービス事業者の手に委ねて良いかどうかの境界線となる。観察したところ、ほとんどのクラウドサービス事業者のシステムでは、重要な特権部分が侵入されると、単一攻撃者は両方の制御権を手に入れることができるようになっていることが分かる。この場合、VM 基盤の側とオブジェクトストレージの側とは別々のクラウドサービス事業者に分離しなければ危険であろうか。理想的には、分離したほうが良い。しかしながら、単一のクラウドサービス事業者内の VM とオブジェクトストレージとの間のデータアクセスは安価に設定されており、複数のクラウドサービス事業者間のデータアクセスは高額に設定されている場合が多い。このような場合で、ゼロトラストの原則をあくまで貫くと、データアクセス料金が比較的高額になってしまうと予想される。他方で、単一の攻撃者が単一のクラウドサービス事業者上の VM 基盤とオブジェクトストレージ基盤の両方に侵入できたとして、その攻撃者が、相当の努力をもって、VM 基盤内の VM インスタンスのメモリ内容に特権命令を用いて読み取りアクセスし、CPU が処理している途中のユーザーのプログラムの暗号鍵を取り出すには、かなりのコストがかかる。なぜならば、その攻撃者は、VM 上で動作しているユーザープログラムの意味的理解 (リバースエンジニアリング) をしなければ、メモリ上のいずれの番地にいずれのタイミングで秘密鍵が読み出さ

れているのかを知ることが不可能であるためである。整理すると、攻撃者は、オブジェクトストレージ基盤そのものに対するアクセスやサーバー側暗号化の鍵を奪取すること (前記 (2-a)、(2-b) に対する攻撃) は、オブジェクトストレージ基盤全体でその格納番地が同じであるので、コスト効率低く容易に可能であるが、VM 上でユーザー自らクライアント側暗号化を施している状態 (前記 (1) のセキュリティ対策) を行なっている場合にその秘密鍵を読み出すには相当の努力を強いられるのである。このように、後者においては、ユーザーによる自らの暗号化処理の実装は、たとえその CPU が単一のクラウドサービス上で動作していたとしても、攻撃者は、相当程度のハードルを突き付けられるのである。これにより、完全なリスクを排除したとはいえなくとも、リスクはそれなりに緩和されているとみることができる。この程度であれば、取扱う機密情報の性質がそれほど機微なものでなければ、ゼロトラストの原則である「暗黙的 Trusted Zone は "可能な限り" 最小化しなければならない。」の "可能な限り" を達成できていると主張することができそうである。

このように、ゼロトラストの原則と、これを貫徹する場合にかかるコストとの比較衡量を行なった上で、いまいちど整理をすると、情報の機密性を 3 段階に分類した上で、それぞれの機密性に基づき、以下のように暗号化の手法を施すことを条件として、パブリッククラウドのストレージサービスを秘密情報の保管のために利用することができるということになるであろう。

(ア) **機密性の保護がほとんど不要な情報 ― 公開予定データ**
クラウドストレージには、クライアント側暗号化なしでアップロードしてよい。

(イ) **機密性の保護が一定程度必要な情報 ― 財務帳簿、入札情報、非公開会議の議事録等、動画・写真等の大容量のデータ**
クラウドストレージには、クライアント側暗号化を施してアップロードしなければならないが、そのクライアント側での暗号化処理 (すなわち Web アプリ等に組み込まれるライブラリを用いた暗号化処理) は、クラウドストレージを提供する者と同一のパブリッククラウド事業者が提供

する IaaS 基盤または PaaS 基盤における CPU とメモリによって行なうことが許容され得る。よって、Web アプリ等を動作させるサーバー VM と、ストレージサービスとは、同一のパブリッククラウド事業者によって提供されていてもよい。これにより、データ転送料金を節約できる。

(ウ) **機密性の保護が最大限必要な情報 ― 個人情報、国家試験の問題文、監視カメラ記録、秘密の公安施設の設計図面、特定機密**

クラウドストレージには、クライアント側暗号化を施してアップロードしなければならない。さらに、そのクライアント側での暗号化処理 (すなわち Web アプリ等に組み込まれるライブラリを用いた暗号化処理) は、クラウドストレージを提供する者と同一のパブリッククラウド事業者が提供する IaaS 基盤または PaaS 基盤における CPU とメモリによって処理されていてはならない。価値がある機密情報であれば、単一のクラウドサービス事業者の特権システムを掌握した攻撃者は、時間をかけて両方の情報を盗み出すことができ、復号化が可能となってしまうためである。よって、Web アプリ等を動作させるサーバー VM と、ストレージサービスとは、異なるパブリッククラウド事業者によって提供される状態とし、クラウドサービス間のネットワークを通じてアクセスされる。データ転送料金はかかるが、保護すべき情報の機密性からすれば、その程度のデータ転送料の増加は許容範囲内である。

(2) 端末・OS の統合管理のシステムは本質的危険（わずかな誤りによる日本全体の行政職員の端末停止のリスク）を有するので、一極集中・単一システム依存を避け、組織間で安全に分散させる必要がある

クラウド上の大規模単一の SaaS 型の OS・端末統制管理システムの業務用端末への一極集中的・全面的導入は、絶対に避けるべきである。たとえ OS 開発メーカーの提供するサービスであっても、避けるべきである。

この手のシステムは、極めて危険で広範囲なサイバーテロを引き起こすための標的となる。当該センターシステムを開発・運用する特権プログラマの過失による瑕

疵 (バグ、セキュリティホール) は、必ず、存在する。いかなるセキュリティシステムにも欠陥がある。欠陥を原因として攻撃者が当該センターシステムを乗っ取った場合、全端末の停止・ロック・強制初期化・ローカル暗号ディスクの解読不能等が一斉に発生する。2030 年頃に、日本と対立する外国勢力がこれを行なった場合を想定する。高度なハッカーは、単独で、あるいはサービス提供者の内部犯行者と通謀して、そのセンターサーバーに侵入したり、設定を変更したりできる。朝になって、すべての国・地方自治体の職員の全端末が一斉にロックされ動かなくなり、データがアクセス不能になり、復旧に 3 ヶ月要し、あるいはソフトウェア自動展開サービスによって標的型マルウェアが自動展開され全部乗っ取られた、というようなことが、起こり得る。これが日本の行政部門で発生し、機能不全に陥っただけで、日本の統治機構は麻痺する。わずかこれだけでも、日本国に多大なダメージを与えることができる。さらに、上記のサイバー攻撃を前置した外国勢力が、日本の行政端末の麻痺の隙を突いて、日本に対してテロ攻撃や物理的攻撃を仕掛けると、日本の統治機構は、もはや回復不能な程度に損傷してしまう。

このような、攻撃者が、狙った国の行政機関全部の端末を一斉に停止させることができるというリスクは、従来の OS やアプリの開発元の自動アップデートのインフラを突いてマルウェア的なソフトウェアを OS やアプリのアップデートに紛れさせて配布することにより大規模な乗っ取りが発生するリスクと比べて、遥かに危険である。全世界的 OS のソフトウェアアップデートのインフラは、確かに悪用され得る。しかし、ここでは選択的に特定のユーザー集団を狙うことはできない。アップデート配信サーバー側では、すべてのユーザーに同じアップデートを配信する。したがって、攻撃者が OS 開発メーカーのアップロードサーバーを掌握しても、特定の種類のユーザーを識別して攻撃することは困難である。そして、アップデートは一斉に起こらない。1 週間くらいかけてランダムな時間に徐々に進行する。発見されたならば、アップデートを停止すればよいので、取り返しがつく。加えて、LGWAN のようなインターネットと遮断された閉域システムでは、アップデートはアップデート専用サーバーを経由して間接的に管理者によって行なわれる。したがって、従来は、OS 開発メーカーが攻撃者に掌握されても、リスクは限

定的であった。

ところが、クラウド上の大規模単一の SaaS 型の OS・端末統制管理システムは、センターサーバー側で、特定の組織端末を指定して特定の指令を与えるだけで、その端末を瞬時にコントロールできてしまう。たとえば午前 10 時丁度に日本国の約 1800 個の行政組織すべてを狙って端末を麻痺させるというような具合で、瞬時にすべてがロック、初期化されるということがあり得る。スマートフォンの遠隔ロック・初期化機能を有効にしていると、スマートフォンを落としたときに遠隔からクラウドサービスにログインし、瞬時にロックしたり、初期化してデータを消したりすることができる。仮にそのクラウドサービスに瑕疵があり、攻撃者がこれを掌握したならば、世界中の同一のシステムを用いているスマートフォンを一瞬にして全部消去できる。それをイメージするとよい。スマートフォンであれば、大したデータは入っていないから、それほど困らない。また再設定すればよい。しかし、日本国の国・自治体の行政庁の端末でこれが起こると、大変なことになる。加えて、ソフトウェア自動配信サービスにより、特定の組織の端末全部で、任意のプログラムを動かすこともできてしまう。

このように、OS・端末統制管理システムというものは、日常的には低水準・中水準のリテラシを有するユーザーの誤用を防ぐことができるメリットがあるが、逆に、その管理システムの中枢部分が乗っ取られたら、組織のすべての端末のセキュリティを侵害できるという点で、本質的に重大な危険性を有している。それでは、OS・端末統制管理システムは利用してはならないのであろうか。必ずしも、そういう訳ではない。危険が緩和できれば、利益のほうが大きくなるから、利用することが正当化できる。従来は、異なるシステムが、オンプレミスまたはクラウドの IaaS 基盤上に分散型導入がなされていたので、そういった危険は緩和されてきた。

従来より、国や自治体で多数のユーザー端末を管理するため、統合管理のシステムが利用されてきた。特に自治体システムでは、各自治体が、自らインターネットと完全隔離された LGWAN 接続系においてこの種の統合管理のシステムのセンターサーバーを構築運用していた。これらのシステムは、オンプレミスのサーバー上に、または、IaaS としてのクラウド上に、組織単位で構築されてきた。これら

のシステムは、多様なものが組織間で分散して利用されており、単独で動作している限り、安全である。そういったシステムの提供企業等が管理する何らかの外部システムによってリモートコントロールされるような仕組みがない限り、安全である。仮に 1 つのシステムに欠陥があっても、影響を受けるのはその組織のみである。仮に 1 つのセンターホストに侵入されても、影響範囲は限定的である。たとえば日本の各省庁、各自治体が何らかの端末統制管理システムをそれぞれ導入しているとする。それぞれが依存する共通的中央部分がない限り、日本全体の行政機能を麻痺させるためには、攻撃者は最大 1800 組織の当該管理システムに対して攻撃を仕掛ける必要がある。しかし、すべての組織のシステムを攻撃者が網羅することは不可能である。そして、同時に攻撃することはできない。このように、従来は OS・端末統制管理システムの危険性は、そういったシステムの組織相互の分散、独自性のある運用、インターネットとの隔離によって、十分な程度に緩和されてきたといえる。

ところが、クラウド上の大規模 SaaS 型の OS・端末統制管理システムについて、単一のシステムを、国の多数の組織の全端末に、また、多数の地方自治体の組織の全端末に導入するとなると、その危険性は極めて高い水準に突然浮上する。その危険性は、たとえ、統制管理システムの管理権限を一極集中させずに、A 省のものは A 省の管理者だけが、B 市のものは B 市の管理者だけがそれぞれ管理するように運用していたとしても、変わらない。なぜならば、本件の危険性は各ユーザー組織の管理者の管理権限を攻撃者に乗っ取られる点にあるのではないからである。本件の危険性は、単一の企業が単一の SaaS サービスとして稼働させる巨大でスケーラブルな統制管理システムのセンターシステムそのものに本質的に内在する技術上の性質によって引き起こされる。これらの巨大な単一システムに 1 箇所でもひび割れが生じたら、攻撃者は 1 のコストで全組織の全端末の管理権限を掌握できてしまう。これにより生じる影響は、前記のとおり、国の統治が直ちにおびやかされる程度に重大なものとなる。その損害リスクは、国民の生命・身体の安全に直結するものといえる。

他方、これを防ぐための方策は、とても簡単である。すなわち、従来と同様、国

の各省庁、各自治体などの組織ごとに、OS・端末統制管理システムを構築運営し、それぞれのセンターサーバーは、異なる管理体系により、それぞれの組織の責任者の主体性のもと運用されることである。従来の安全な仕組みの延長線上で拡大させることである。OS・端末統制管理システムのクラウド化を推進することは、良いことである。ただし、それは IaaS 型のクラウドを用いる場合に限られる。このようなクラウドを用いた安全な OS・端末統制管理システムの導入と、クラウド上の大規模 SaaS 型の OS・端末統制管理システムの業務用端末への一極集中的・全面的導入とは、危険性の次元が全くことなる。前者の危険は、対処可能であるが、後者は、現在の技術水準では、対処不能である。

(3) 重要な業務システムで用いる ID 認証基盤の構築のためにクラウドサービスを利用する場合は、ソフトウェアそのものに対する衆人環視が確保されているソフトウェアを IaaS で用いること（SaaS の ID 認証基盤の利用は、可能な限り避けること）

ゼロトラスト原則を実現するためには、アクセス元ネットワークにかかわらず、すべての接続要求について、ユーザー認証を施すことが必要である。これはもともと社内 LAN などの多数のユーザーがアクセスする可能性がある環境では当然のことであるが、従来は社内 LAN の閉域性を過信していたので、社内 LAN に到達できた攻撃者は、認証のないファイルサーバーの情報に自由にアクセスできた。そこで、各サーバーに認証・認可・ログ記録を施すことになるが、これは、たいていの従来の各サーバーで実現できる。Web アプリについても同様である。ただ、サーバーごとにユーザー管理を個別に行なうと大変なので、ID 認証基盤を用いて、ユーザー管理やアクセス権限の範囲の設定を一元化することが便利である。

ID 認証基盤は、オンプレミスで構築するのがもっとも安全である。それが困難な場合は、クラウドで構築することも、十分な対策をとれば、許容されるであろう。一般的な OS 上で動作する、さまざまな製品やオープンソースソフトウェアが存在する。しかし、ここで注意しなければならない点が 1 つある。クラウドで構築する場合には、PaaS や SaaS による ID 認証サービスを利用して ID 認証基盤を実現するべきできはない。PaaS や SaaS による ID 認証基盤の実現とその利

用は、ゼロトラスト原則に逆行する。IaaS (VM 基盤) を用いて、ユーザー自らの支配・管理下において、ID 認証基盤をインストールし、構築し、運用する必要がある。

その理由は、次のとおりである。ゼロトラスト原則においては、Trusted Zone を最小化しなければならない。そして、Untrusted Zone と Trusted Zone との間のサーバーや Web アプリには、必ずユーザー認証を施さなければならない。そのサーバーや Web アプリの VM 環境やプロセスそのものを支配・管理する特権を有するすべての構成要素は、Trusted Zone に該当する。これらのサーバーは ID 認証基盤を参照し、ユーザー認証はすべて ID 認証基盤の認証結果を信頼することとなる。よって、ID 認証基盤そのものは、ゼロトラスト原則における Trusted Zone に該当する。

ID 認証基盤そのものが Trusted Zone に該当するならば、Trusted Zone には、ユーザーが可能な範囲で最大限のセキュリティ対策を施し、集中的にこれを保護しなければならない。ゼロトラスト原則においては、従来モデルと比較して、Trusted Zone に要求されるセキュリティレベルは可能な限り高められる必要がある。これは、Trusted Zone 以外はすべて信頼できないものとみなして、Trusted Zone を可能な限り極小化するというゼロトラスト原則から導出される要求である。ID 認証基盤はゼロトラスト原則における Trusted Zone のうちでも、特に高いセキュリティレベルが要求される。

そこで、ID 認証基盤そのもののソフトウェアの脆弱性の存在が問題となる。ID 認証基盤には、ソフトウェアそのもの (バイナリのみでも良い) が公開・配布されており、これをユーザーが自らのオンプレミスまたはクラウドの IaaS 上で動作させることができるものと、ソフトウェアそのものが非公開のものの 2 つに分かれる。少なくともバイナリが公開されているソフトウェアで、多数のユーザーによって利用されているものは、多数のユーザーによるリバースエンジニアリングによる衆人環視的なセキュリティ検証が施されている。そして、多数の脆弱性が発見され、すでに修正されている。技術的用語でいうと、「枯れている」状態である。このような多数のユーザーによるセキュリティ検証と指摘のみが、高度複雑なソフト

ウェアの脆弱が発見され、それが修正されることの確立を、サイバー攻撃耐性を得るために必要な水準に高めるために、現在発見されている唯一の手法である。したがって、ID 認証基盤ソフトウェアを自前サーバーやクラウドの VM 上で動作させ、その環境はユーザー自らが完全に支配・管理している場合、可能な限りのセキュリティが実現できることとなる。

一方で、ID 認証基盤そのもののソフトウェアがバイナリを含めて全く非公開で、クラウド上のサービス (SaaS や PaaS サービス) としてのみ利用可能な類型のソフトウェアサービスが存在する。これは、原則として利用すべきではない。そのようなクラウドサービスを提供する事業者が動作させているソフトウェアの安全性を多数のユーザーが衆人環視によって検証しようにも、ソフトウェアのバイナリが非公開であるため、検証することができない。ソフトウェアの脆弱性は、できればソースコード、ソースコードが非公開であればバイナリをリバースエンジニアリングして、はじめて発見し、指摘し、修正させることが可能である。これが不可能な ID 認証基盤の非公開サービスは、検証をする手段がない。開発元自らの限られた人数のエンジニアがこれを検証し、または特定少数の外注業者に検査させ、潜在的脆弱性を探すことによっては、従来と同等の市場水準のセキュリティを実現することは、不可能である。したがって、クラウド上のサービス (SaaS や PaaS サービス) としてのみ利用可能な ID 認証基盤のソフトウェアには、重大な脆弱性が発見されずに存在している可能性が極めて高い。サービスとしての ID 認証基盤のソフトウェアの特権領域は、極めて広い。多数のユーザー企業が、クラウドサービスとしての単一の ID 認証基盤サービスを利用する場合、多数のユーザー企業すべてで、その ID 認証基盤サービスの Trusted Zone が論理的に横つなぎされた状態となってしまう。もちろん、ユーザー企業からは、通常は他のユーザー企業の情報にはアクセスできない。しかし、この Trusted Zone の論理分割は、とても弱い隔離で成り立っている。ID 認証基盤サービスのプログラマがプログラムによって実装した論理分割に過ぎない。ID 認証基盤サービスのソフトウェアや、あるいは、これを動作させているより下位の VM 基盤やストレージ基盤のレイヤを攻撃者が掌握した場合、同じ ID 認証基盤サービスを利用しているすべての企業のす

べてのユーザーオブジェクトがアクセスできるすべてのリソースが、直ちに危険にさらされる。すなわち、極めて多数の企業がクラウドサービスとしての単一の ID 認証基盤サービスを利用している場合、そのサービスの内部実装領域としての Trusted Zone は、極めて広い。これを信頼することは、Trusted Zone を極小化するべきというゼロトラスト原則に反する。

他方で、IaaS 上で動作可能なバイナリ公開型の ID 認証基盤ソフトウェア製品は、Trusted Zone の極小化の点でも、多様性の点でも、安全である。仮に IaaS のクラウド VM 上で個別にユーザーが ID 認証基盤ソフトウェアをインストールして稼働させている場合で、かつ、そのインストール方法に多様性がある場合、攻撃者は 1 つの組織ごとに 1 回ずつ攻撃を仕掛ける必要があるが、これはとても難易度が高い。クラウドサービスとしての単一の ID 認証基盤サービスを利用する場合、攻撃者は 1 回の攻撃ですべてのユーザー企業のセキュリティを同時に侵害することができる。IaaS 上で独自にインストールして運用する場合、そのような攻撃に対して安全性が極めて高くなる。クラウドサービスとしての ID 認証基盤の場合は、たとえ単一の企業を狙った標的型攻撃であっても、その攻撃を成功させる手段としてその ID 認証基盤サービスの基盤環境そのものを攻撃者が侵害することに精巧した場合、クラウドサービスとしての同じ ID 認証基盤サービスを利用しているすべてのユーザー企業が影響を受けてしまう。ID 認証基盤サービスは、ゼロトラスト原則における最重要部分の Trusted Zone である。セキュリティ侵害が発生したとき、他の Trusted Zone すべてに対する自由なアクセスを許可してしまう、最も危険な、最も重要な Trusted Zone である。そのような最重要領域を、IaaS で動作可能な公開型ソフトウェアを利用するよりも脆弱性が存在する可能性が高く、かつ、単一の攻撃ですべてのユーザーが侵害され得る可能性があるクラウドサービスとしての単一の ID 認証基盤サービスに委ねてはならない。

このような理由で、ID 認証基盤をクラウド上で動作させる場合、前述のとおり IaaS のクラウド VM 上で動作可能な、すなわちソフトウェアのバイナリがユーザーに公開されているものを利用しなければならない。加えて、他の種類の IaaS の VM と比較して、さらなるセキュリティ対策を施すべきである。すなわち、攻

撃者は IaaS の VM を動作させるクラウド事業者の基盤環境 (VM 基盤、ストレージ基盤) を侵害し得るが、その場合にも攻撃者が容易に ID 認証基盤の構成情報を取り出したり、改ざんしたりできないように、いくつかの対策を立てる必要がある。最も安全な方法は、その IaaS をマルチクラウド環境で構築し、VM (CPU、メモリ) ユーザー向けネットワークはクラウド事業者 A、ストレージはクラウド事業者 B を利用し、クラウド事業者 A の VM (CPU、メモリ) が、ローカル暗号化の手法を用いて、クラウド事業者 B 上の仮想ディスクを読み書きする方法である。これは構築時には若干の手間がかかるが、運用コストは低い。なぜならば、クラウド事業者 A とクラウド事業者 B との間で発生する通信量が課金されるが、ID 認証基盤は、ファイルサーバー等と比較して、ディスク I/O の量は極めて少なく、最も高額なネットワーク課金 (帯域幅に応じて課金される) は安価に抑えられるためである。もちろん、クラウド事業者 A とクラウド事業者 B の両方のインフラが動作していなければ、システムは動作しない。いずれかがダウンすると、システムがダウンしてしまう。単純にダウンの頻度は 2 倍となる。しかし、ID 認証基盤は複数台のサーバーを運用することが可能である。ここで、各クラウド事業者のアベイビリティゾーンの仕組みが活かされる。異なる組み合わせで、VM (CPU、メモリ) ユーザー向けネットワークと、ストレージとの対応関係を作っておけば、すべてが同時にダウンする可能性は極めて低くなる。

このようなクラウドを最大限に活用した安全な ID 認証基盤を構築することが、ゼロトラスト原則に基づく Trusted Zone を "可能な限り" 極小化するという基本方針を実現する唯一の方法である。そして、この方法におけるマルチクラウド的な安全構築方法は、パブリッククラウドを利用しているにもかかわらず、サイバー攻撃者から可能な限り安全に保護される。クラウド事業者 A の基礎部分を掌握したサイバー攻撃者であっても、クラウド事業者 B の基礎部分を掌握しなければ、構成情報の奪取や書き換えが極めて困難である。動作中の VM 環境のメモリへの介入が必要であるためである。これは原理的には可能であるが、攻撃コストが極めて高く長時間を要し、その途中で検出され得る。

いかなる場合でも、異なる組織間で ID 認証基盤のインスタンスや管理権限を

共有することは、避けなければならない。ID 認証基盤に対する管理権限は極めて強力であり、ここが奪取されたり、技術的権力者が暴走したりすると、極めて危険である。異なる法人 (地方自治体同士は異なる法人であり、国と地方自治体も異なる法人である) や組織 (各省庁) 間で、ID 認証基盤は、明確に互いに分離しなければならない。特権的管理者が出現して、複数の組織に渡って横断的な管理権限を行使できるような状態となってはならない。ID 認証基盤は、多様性を重視し、個別に構築されるべきであるが、たとえ同一の ID 認証基盤ソフトウェアを用いる場合でも、IaaS のレイヤで互いに分離されており、かつ、各 IaaS の VM は異なるクラウド管理領域に属している必要がある。

[redacted]において、「国 (ID 基盤の整備)」として「全政府機関横断的に ID 認証基盤を利用する。」、「自治体 (ID 認証基盤の整備)」として「各地方自治体において、ID 認証基盤を利用する。」との記載があるが、この方針の表現は、「ID 認証基盤を利用する。」から、「ID 認証基盤を構築する。」に改める必要がある。すなわち、「利用する。」という表現は、あたかも単一のいずれかの主体 (デジタル庁であろうか) が ID 認証基盤を構築し、それを各省庁・各地方自治体が自らのシステムにおけるユーザー認証のために「利用する」というように誤読されてしまうリスクがある。これでは、多様性と管理権限の分散によるセキュリティを実現することができない。ゼロトラスト原則における Trusted Zone の "可能な限り" の極小化を実現することができない。ID 認証基盤は、各組織ごとに異なるものでなければならないし、単一の製品に依存してはならない。単一化と単一製品への依存は、セキュリティ上の甚大な被害の原因となる。国の全統治機構が、高度なサイバー攻撃者や、悪意を持った管理権限者により、一瞬で侵害される原因となる。それは、避けなければならない。

(4) 重要なデータベースはクラウド上では SaaS、PaaS ではなく IaaS で構築しなければならない

データベースのデータは、ゼロトラスト原則における Trusted Zone の最たるものであり、特に機密性、完全性、可用性が要求される領域である。クラウドを用いてデータベースを構築・運用する場合、① クラウドサービスとしてのデータベ

ース (SaaS、PaaS) を利用するパターンと、② IaaS の VM 上で自らデータベースソフトウェアを稼働させるパターンの 2 種類が存在する。

ここで、機密性、完全性、可用性のいずれかを要求されるデータを扱う場合は、① クラウドサービスとしてのデータベース (SaaS、PaaS) を利用することは避けなければならない。その理由は、データの漏えい、損傷、破損、またはパフォーマンス劣化が発生した場合における責任の所在があいまいになるためである。クラウドサービスとしてのデータベース (SaaS、PaaS) では、そのソフトウェアのバイナリを含めたバージョン管理や内容の管理、すなわち挙動に対する完全な支配権がユーザーにはなく、クラウド事業者のプログラマやオペレータの側にその支配権が移行する。そして、データベースのプログラムはすべてのユーザーで同じものが実行される。加えて、データそのものも、ユーザー間の境界は薄く、ソフトウェアによる一枚の壁を隔てた論理分割によって分離されている。この状態でソフトウェアに脆弱性が存在したり、何らかの欠陥が存在した場合、すべてのユーザーのすべてのデータに攻撃者がアクセス可能となり、甚大な機密性の損害が発生する。加えて、データが少しずつ壊れていくような現象、性能劣化、データ検索結果の欠落の原因究明も困難となる。なぜならば、このような問題解決は、クラウドサービス事業者の秘匿領域のソフトウェアコードの問題を研究するため、クラウドサービス事業者の従業員 (特に、そのようなソフトウェアコードを理解できる技術者) に頼らないといけないが、仮にクラウドソフトウェア側に何らかの問題があると、このような問題は、多数のユーザーで同時多発的に発生する。すると、クラウドサービス事業者側の高レベルの従業員がいよいよ対応してくれるまで、とてつもない時間がかかる。その間に機密性、可用性、完全性が次々に失われていく。

これと比較して、② IaaS の VM 上で自らデータベースを動作させる場合、ファイルシステムを構成するストレージのブロックレベルでの完全性が保証されている限り、自らの手で必ず原因究明が可能であるし、問題が発生することを予防することも容易である。そして、ストレージのブロックレベルでの不具合は、さまざまなログや検査結果により、かなり確実に知ることができる。

加えて、クラウドサービスとしてのデータベース (SaaS、PaaS) を利用する場

合、その機能や挙動や API、コマンドがパブリッククラウド事業者ごとに互いに異なるので、ベンダロックインの危険が生じる。また、パブリッククラウド事業者側のプログラマの嗜好によって突然挙動が変わってしまうリスクもある。よって、重要なシステムについては、クラウドサービスとしてのデータベース (SaaS、PaaS) を利用すべきでない。欠陥があった場合の損害が、IaaS 上でのデータベース自前構築をする場合と比較した管理の楽さの利益を上回るためである。

クラウドサービスとしてのデータベース (SaaS、PaaS) を利用する利点として、データ損傷があった場合にもクラウドサービス事業者の従業員によってデータを復旧してもらえる可能性があり、また、クラウドサービス事業者の側のプログラムに瑕疵があった場合に損害賠償を受けられるという点が主張されることがある。しかしながら、実際にデータが損傷した場合、ユーザー側に責任がなく、クラウドサービス事業者側に責任があることを立証することは、容易ではない。そして、仮に破損の立証に成功したとしても、結局、損害賠償金額は SLA によりわずかな金額に制限されてしまう。よって、この利点は実際にはあまり役に立たず、少なくとも、IaaS 上でのデータベース自前構築をする場合に得られるメリットを上回るとはいい難い。

ただし、例外的に、Web サイトでのお問い合わせページの個人情報などの少量のもの、アンケート集計、損傷しても良いような膨大な量のログの記録 (ただし、ログに認証クレデンシャルや Cookie などの秘密情報が含まれてはならないことはもちろんである) などのジャンクデータ的なものを、ビッグデータ的に解析するためには、クラウドサービスとしてのデータベース (SaaS、PaaS) を利用しても、欠陥があった場合の損害が少なく、利用利益が上回るため、このような場合については、利用しても良いと考えられる。

(5) HTTPS 中間者攻撃型プロキシサービスをやむを得ず構築する場合は、オンプレミス型または IaaS 型を利用すること

最近、ゼロトラストと称して、HTTPS 中間者攻撃型プロキシサービスを導入するパターンが散見される。HTTPS 中間者攻撃型プロキシサービスとは、ユーザー端末とインターネット上の Web サーバーとの間で折角確立される安全な

HTTPS 通信に対して、中間者攻撃と呼ばれる一種のサイバー攻撃を加え、その HTTPS 通信の内容をいったん解読し、再暗号化する仕組みにより、HTTPS の内部を流れる平文を記録したり、アクセス先の URL に基づいてアクセスを許可・拒否する、ファイアウォール的なサービスである。

このような HTTPS 中間者攻撃型プロキシサービスは、ゼロトラストとは全く関係がない。ゼロトラストと称してこれを販売する事業者が時々見られるが、ゼロトラスト原則に合致しないばかりか、むしろ逆行している。すなわち、ゼロトラスト原則においては、ネットワークの構成要素を一切信用することなく、ユーザー端末とアクセス先のシステムとの間でエンド・ツー・エンドの暗号トンネル (HTTPS 等) を構築し、安全な通信を実現することが重要である。この際、たとえば Web ブラウザ等の HTTPS クライアントと、サーバーとの間で、Trusted Zone に一時的にアクセスが可能なセキュアチャネルが張られることになる。いわば、その暗号チャネルは、一時的に、Trusted Zone の一部となるのである。ところが、その HTTPS の暗号を HTTPS 中間者攻撃型プロキシサービスを用いて解読するとなると、当該 HTTPS 中間者攻撃型プロキシサービスは、その Trusted Zone を流れるデータを平文として読み取ることが可能となる。ここで、もともとのユーザーとサーバーとの間の HTTPS 通信がユーザー認証されているか否かは、SSL トンネル内を流れる Cookie のセッションキーや認証サーバーから返されるクレデンシャル証明情報 (たとえば、署名されたタイムスタンプ付きの認証符合) によって識別される。仮にこれらの SSL トンネル内を流れる秘密のセッションキー等が攻撃者に漏えいしたら、ゼロトラストモデルの場合、攻撃者は直ちに自らの PC からセッションを乗っ取ることができてしまう。社内 LAN の内側とインターネットとを峻別している場合、攻撃者は、何らかの方法でセッションキーを入手しても、別の手順としてその内部 LAN に侵入する必要があり、そうしなければサーバーにアクセスできなかったので、このようなリスクは低い。しかし、完全にすべてのサーバーをインターネットに移行することを理想とするゼロトラストモデルの場合、キーが奪取されたら、もうそれで終わりである。攻撃者はインターネットを経由して、堂々とそのキーで認証済みセッションを騙ることができてしまう。よって、

HTTPS 中間者攻撃型プロキシサービスそのものは、Trusted Zone の一部として、極めて高いセキュリティ上の保護を必要とする。加えて、HTTPS 中間者攻撃型プロキシサービスはゼロトラストモデルの実現には不必要で無関係なものである。これは、単に従業員が業務時間中に不必要な Web サイトにアクセスしていないかどうかを監視するためのツールである。HTTPS 中間者攻撃型プロキシサービスは、不必要に Trusted Zone を広げる。よって、HTTPS 中間者攻撃型プロキシサービスの存在は、"可能な限り" 暗黙的 Trusted Zone を極小化することを求めるゼロトラスト原則に反する。

それでもなお、セキュリティが低下することを承知の上で、何らかの正当な理由があり、HTTPS 中間者攻撃型プロキシサービスを運用したい場合、HTTPS 中間者攻撃型プロキシサービスのインスタンス (すなわち、プロキシサーバー) は、オンプレミス型または IaaS 型を用いなければならない。SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスは、決して利用してはならない。その理由は、HTTPS 中間者攻撃型プロキシサービスのソフトウェアにおいて SaaS / PaaS 型を利用すると、当該 HTTPS 中間者攻撃型プロキシサービスのソフトウェアにおける脆弱性が 1 つでもあれば、攻撃者は、すべてのユーザー組織 (他の契約者を含む) のすべての通信内容をすべてキャプチャすることが可能となってしまうためである。サイバー攻撃者が、組織のユーザーの HTTPS 通信をキャプチャできてしまうことの脅威は、最大限に危険なものである。なぜならば、HTTPS 通信は前記のとおり認証情報をやりとりし、セッションキーをその内部で Cookie 文字列として常に授受するが、攻撃者はこれらをすべてキャプチャ可能となる。HTTPS を用いるすべて Web メールの送受信の本文および添付ファイルも、すべての重要なシステムのための Authenticator アプリのワンタイムパスワードの初期認証時の QR コードの平文データも (これを攻撃者に奪取された場合、絶望的な結果が発生する)、メッセンジャーや Web 会議システムで送受信されるチャット、映像、音声、ファイルも、すべてキャプチャ可能となる。仮に企業システムのシステム管理者の通信がキャプチャされれば、攻撃者は、すべてのシステムにログインでき、すべての管理特権を用いて、ユーザー組織に対して破壊的なインパクトのある

攻撃を仕掛けることができる地位を奪取できる。ワンタイムパスワードも、すべて無力となる。ワンタイムパスワードのアプリの初期設定時に読み込ませる QR コードの画像ファイルが HTTPS 中間者攻撃型プロキシサービスで一度復号化されている状態を攻撃者はキャプチャできるためである。このような危険があるにものかかわらず、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスを構成するソフトウェアは、そのバイナリが未公開なので、衆人環視によるリバースエンジニアリングに基づく脆弱性の発見・修正勧告が全く不可能である。すなわちサイバー攻撃者としては、いつまでも修正されない脆弱性を活用して、クラウド上の当該サービスに集中して通信を行なう多数の企業の多数のユーザーを選択的または網羅的にキャプチャし放題ということになる。このような SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスの脆弱性によるリスクが現実化した場合の破壊力は、たとえてみれば、パブリッククラウド事業者の全ユーザーのストレージ領域が無差別に攻撃者によって 10 年間くらい読み取られていたことが 10 年後に発覚した場合と同程度の破壊力を有する。このような甚大な被害が発生する可能性にもかかわらず、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスの提供業者は、ユーザーに対して、SLA 契約として、わずかな損害賠償金額を上限に設定していることが多い。これを超えたユーザー側の損害は補償されない。この状態では、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスは、通常の企業でも、利用すべきではない。特に、国や地方自治体のシステムは、国の存立に関わる極めて重要なシステムを扱うので、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスは、決して利用してはならない。

それでも、どうしても HTTPS 中間者攻撃型プロキシサービスを利用したい場合は、オンプレミスまたはクラウド上の IaaS 型で動作するソフトウェアで、かつ、広範囲で長年利用されてきており、高度な技術者群によって衆人環視としてのリバースエンジニアリングによる脆弱性検査と公表、修正をひんぱんに経てセキュリティが強化された、定評のある HTTPS 中間者攻撃型プロキシソフトウェアのみを用いるべきである。そのようなソフトウェアは、バイナリすら非公開な SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービス (それらのサービス事業者の

支配領域のみで動作していて、検証できない) と比較して、各段に安全性が高い。そして、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスと比較して、オンプレミスまたはクラウド上の IaaS 環境には、SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスを狙った無差別な攻撃者が入り込む危険が低い。SaaS / PaaS 型の HTTPS 中間者攻撃型プロキシサービスにおいては、1 のユーザー組織を狙った攻撃者は、当該サービス全体を掌握することができるので、これによる巻き添え被害を食らう可能性が高い。極めて多数のユーザー組織で、単一のサーバーを共有しているためである。しかし、自ら HTTPS 中間者攻撃型プロキシサービスをインストールして構築する場合は、自らの組織のみでその一応隔離されたサーバーを利用することになる。攻撃者はそのようなサーバーごとに侵入をする必要があるが、インストール手法や動作させている OS の多様性によって、侵入は極めて高いコストがかかる。これにより、安全性は飛躍的に向上する。

(6) 電子メールサーバーシステムは、従来のオンプレミス同様の多様性による強靭的セキュリティを、クラウド環境でも実現する必要がある

電子メールは、依然として重要なインフラであり、高いセキュリティ (完全性・機密性・可用性) が要求される。高いセキュリティを要する多数の組織の電子メールシステムは、できるだけ異なるメールサーバーソフトウェアによって、多様性を有する構築・運営・管理によって、それぞれ独立・分離されて運営される必要がある。これは、オンプレミスシステムにおいても、これまでにすでにある程度達成されてきている。クラウド時代になっても、これは変わらない。すなわち、クラウドを用いる場合においても、SaaS としての大規模電子メールサーバーシステムを利用するのではなく、クラウド上の IaaS VM とストレージを用いて、各組織ごとにそれぞれ独自の流儀に基づいて電子メールサーバーを運営することが重要である。

SaaS としての大規模電子メールサーバーシステムには、完全性・機密性・可用性に関する重大な問題がある。SaaS としての大規模電子メールサーバーシステムは、大量の異なるユーザー組織を単一のソフトウェアで稼働させている。大規模電子メールサーバーシステムは、ユーザーごとの隔壁をソフトウェアによって作り出している。したがって、大規模電子メールサーバーシステムの管理領域は、Trusted

Zone としての機能を提供する。ところが、大規模電子メールサーバーシステムのコードは、バイナリも含めて一切公開されておらず、多数のユーザーによる衆人環視的なリバースエンジニアリングを含めた脆弱性の発見・指摘・修正勧告行為を経ていない。よって、大規模電子メールサーバーシステムのソフトウェアには、多数の未修正の脆弱性や不具合が存在する。そして、電子メールサーバーシステムの特性上、メール本文は暗号化されていない。この状態において、サイバー攻撃者が、同一の大規模電子メールサーバーシステムでホストされている多数のユーザーのうち 1 ユーザーを狙って当該システムに攻撃を行ない、これが低いレイヤで成功した場合、完全性・機密性・可用性は、すべてのユーザー組織のすべてのユーザーのすべての電子メールに及ぶ。これは甚大な被害をもたらす。特に、最近の SaaS としての大規模電子メールサーバーシステムは、各メールボックスに対して、数十 GB ものデータの保管を可能としている。機密性に対する影響は、過去数十年分の数十 GB ものデータに及ぶ。

したがって、SaaS としての大規模電子メールサーバーシステムは、従来通り、あまり機密性が高くない用途、たとえばインターネットからのあまり重要でない一般的な問い合わせ目的などのために限定して利用すべきである。省庁間、自治体間、または省庁／自治体の間の業務的電子メールの授受は、高いセキュリティが必要である。これは従来 LGWAN 内で完結していた場合も多いが、今後は、インターネットを通じて伝送されることも増える。この場合でも、インターネット上にオンプレミスやクラウド上の IaaS VM を用いて電子メールサーバーとメールボックスを構築するノウハウは、従来の閉域網のオンプレミスのメールサーバーを構築する場合とほとんど変わらない。重要な点は、メールサーバーをインターネットと直接接続する場合、メールサーバーのソフトウェア (多くはオープンソース方式で開発されており、極めて多数のユーザーが衆人環視しているので、脆弱性は十分に発見されており、安全性・安定性が高い) のアップデートを必要に応じて欠かさないこと (ただし、何でも新しいバージョンにアップデートすれば良いというものではない) と、メールサーバーへの接続は SSL 証明書の検証を必須にすること、および、クラウド上のデータはいつでも消える可能性があるため、ローカルにデータのバッ

クアップを取る (定期的にコピーする) ことである。各組織ごとに異なる管理の異なるメールシステムが利用されている場合で、管理権限も組織間で互いに完全に独立分離していれば、セキュリティは飛躍的に向上する。攻撃者がひとたび国・地方全体のメールサーバーを掌握すると、統治上の危険性は極めて増大する。これを防ぐために、上記のような多様性のあるメールサーバーの管理手法を採れば、攻撃者は 1 つ目の組織に苦労して侵入できても、2 つ目の組織への侵入には、また一苦労しなければならない。すべての組織へ単一攻撃で侵入できるマスターキーがないことが、安全のための条件である。

(7) ネットワーク機器、SDN 装置、ファイアウォール等は、多様性が重要であり、世代もファームウェアのバージョンも異なることが望ましく、全自治体に適用するインターネットベース・クラウドベースのネットワーク機器の中央管理システムは避ける必要がある

仮に各省庁・各自治体向けのネットワークの構築を国が実施またはサポートする場合、ネットワーク機器、SDN 装置、ファイアウォール等においては、多様性が極めて重要である。すなわち、単一のソリューションや製品群を、すべての各省庁・各自治体で利用してはならない。これはセキュリティ上壊滅的な結果を生じさせる。

システマティックに単一的方針によって設計されるネットワーク、特に、ネットワーク機器、SDN 装置、ファイアウォール等として単一製品が利用されているネットワークは、1 つでも脆弱性が存在している場合、その同じ脆弱性によって、直ちに全ネットワークが危険にさらされる。攻撃者は、1 つの攻撃手法によって、全体のセキュリティ侵害を引き起こすことができる。これは極めて危険である。この危険を予防するためには、できるだけ多様なネットワーク機器、SDN 装置、ファイアウォール等の装置でネットワークを構成する必要がある。すなわち、組織ごとにできるだけ異なる機器を用いる必要がある。A 自治体、B 省、C 庁それぞれで、互いにできるだけ異なる装置が利用されている必要がある。ただ、優秀な装置のバリエーションには限りがある。単一の装置が複数の組織で利用されることも当然発生する。この場合は、ファームウェアのバージョンの違いと構成 (設定コンフィグ)

の違いが重要となる。ファームウェアのバージョンを、例えば特定のバージョンに全部常に揃えるというような運用は、通常、極めて脆弱な結果となるので、これは絶対に避けるべきである。いかに最新のファームウェアであっても、脆弱性が存在し得る。そのような脆弱性を突く攻撃の成功・失敗は、ファームウェアのバージョンによって微妙に異なるコンパイル結果によって左右される。同一の攻撃コードであっても、ファームウェアのバージョンが異なると、致命的被害を免れる。ファームウェアのバージョンが互いに異なれば、大半の装置は、クラッシュするだけで済むことが多い。だが、ファームウェアのバージョンが全部一緒であれば、全部の装置がサイバー攻撃者によって侵入されることになる。加えて、ネットワーク機器の設定 (コンフィグ) も、それぞれ異なる流儀によって設定されていることが重要である。仮に機器とファームウェアが同一であったとする。コンフィグの流儀が同じであり、管理的・計画的に同じコンフィグを配布するような場合は、すべてのデバイスが同一の攻撃手法で侵入できる。低いコストで全部のネットワーク機器を掌握できる。コンフィグの流儀が異なれば、一方には攻撃できるが、他方には攻撃できないというような安全な状態を維持できる。攻撃者による攻撃コストを高めることができる。

このように、サイバー攻撃に対して免疫力の高いネットワークは、各組織の各機器のベンダもできるだけ異なり、種類ができるだけ異なる状態を維持して、はじめて実現する。ところが、そのような多様性がある強靭なネットワークを意図して中央集権的に構築しようとすると、これは、なかなか大変である。そもそも、互いに独立した多組織にまたがる巨大なネットワークを、中央集権的に構築しようとすることそのものが、多様性を減退させ、サイバー攻撃のリスクを高める。この問題を解決するためには、地方自治体 (本来は、中央省庁も) にまたがるネットワークは、エッジの部分まで国 (デジタル庁) で整備するのではなく、仮にデジタル庁が全体のネットワークの構築を実施またはサポートする場合でも、国 (デジタル庁) はセンター部分のみを構築し、そこに接続するためのプロトコル (そのプロトコル = 規格も、複数存在する必要がある) を定め、エッジ部分からの繋ぎ込みは、各自治体の担当者が自ら行なう (そのための機器も、各自治体が用意する。もっとも、

1Gbps 近くの速度が出るルータは、多様なベンダのものが、いずれも数万円で用意できるので、費用は限定的である) 分散型の方式とするのがよい。これには 2 つの利点がある。第一に、自然にベンダ、機器、ファームウェアのバージョン、コンフィグが異なる状態が形成されるので、多様性が形成され、サイバー攻撃に対して極めて強くなる。第二に、各自治体の IT 職員のセキュリティ能力やネットワーク運営能力を極めて高い水準に向上させることができる。各自治体の IT 職員は、これまで LGWAN のルータは業者任せで構築してもらってきた。LGWAN のルータよりも内側の自治体庁舎内と庁舎間のネットワークのみを、自ら構築してきた。これでは LAN に関する知識しか身に付かない。しかしながら、自ら機器を選定して国・自治体ネットワーク (WAN 部分) への接続部分を構築することを要求され、インターネットとも関連し、かつ、高いセキュリティ水準を求められるようになったならば、WAN に関する知識やセキュリティの知識も身に付ける必要が生じるのである。仮に安全な設定に失敗して脆弱な状態で放置すると、自らの責任となる場合に、初めて、必要な勉強や対策を施そうという意欲が生じる。これにより、各自治体の IT 職員の組織的なセキュリティ能力は、現在よりもかなり向上する。このモデルは長期的なセキュリティを考慮した場合には最良であるが、短期的にみるとリスクを伴うので、十分な緩和策が必要となる。リテラシ能力が低い自治体向けには、いくつかの推奨ベンダの推奨機器を間違えようがない程度のある程度画一的な機器の構成方法のガイドラインとともに列挙しておくのである。リテラシ能力が低い自治体の職員は、まず、それを用いて自らの WAN の接続を実施する。うまくいけば、徐々に勉強を重ねて、より高機能・高性能な機器を自ら選び、さまざまな付加的機能も利用し、だんだんと WAN 部分を拡充していくであろう。そのうちに、自らよりセキュアな機器を自作する自治体が登場する可能性もある。このような良い流れの傾向が続けば、日本全体の自治体のセキュリティ能力は飛躍的に高まる。このような手法における要点は、1 つの自治体 (または省庁) のネットワーク機器の設定の瑕疵によって、その 1 つの自治体 (または省庁) のセキュリティが侵害された場合、他の自治体 (または省庁) や国・地方全体のシステムにそのセキュリティ侵害が拡大していくことを予防する点にある。ゼロトラスト原則に従う限

り、社内 LAN は Untrusted Zone とみなされるので、ネットワークが侵害されても致命的ダメージは生じないかも知れない。しかし、ゼロトラスト原則に適合するように庁舎内 LAN のサーバー群をすべて安全な状態にするまでには、時間を要するし、いつまでもそのような移行は完了しないかも知れない。当面の間、省庁内ネットワークのサーバーや、自治体内ネットワークのサーバーは、ネットワークそのものを一応安全なものとみなす伝統的モデルのもと稼働し続ける必要がある。そのため、一の自治体へのネットワーク侵害によって他が侵害されることは、避ける必要がある。そのためには、現在の LGWAN と同様に、国・自治体間の完全な自由な通信を許容することなく、特定の許可した通信のみを許容するような仕組みが、ネットワーク中央側において、過渡期的に必要である。

いずれにしても、最も避けるべき手法は、上記のような多様性を喪失させる手法である。全自治体において、単一のネットワーク装置を用いて、同一のファームウェアのバージョンに統一し、インターネットベースの中央的管理システムを利用する方法である。これは極めて危険である。多様性がなく免疫がないので、サイバー攻撃に対して極めて弱くなる。各装置に対する攻撃だけでなく、中央の管理システムに対する攻撃に対しても、とても脆弱となる。中央の管理システムが掌握されると、全ネットワーク機器のデータを消去することができ、復旧まで超長時間を要し、大混乱に陥る。各自治体、各省庁のネットワーク機器の認証クレデンシャル (パスワードや、SSH 認証の公開鍵) は、それぞれ異なることが極めて重要で、それらの機器には、それらの互いに異なる認証クレデンシャル以外では、特権的にログイン・制御することができないことが原則的に重要である。中央管理システム型では、その原則が失われる。ネットワークをシステムとしてみた場合、各ネットワーク機器の管理領域は、紛れもなく、Trusted Zone である。ゼロトラスト原則における Trusted Zone の極小化の原則により、すべてのネットワーク機器に対する特権を中央の管理システムによって与えるべきではない。

(8) クラウドを用いた AI サービスの利用については、AI サービス提供事業者からの偏向攻撃に対する安全措置を講じること

業務システムにおける AI システムの利用において、クラウドを利用する場合、

IaaS で自らデータやソフトウェアを含めた排他的支配・制御を確保できる場合にのみ使用する。他人が開発したソフトウェア、データセット、アルゴリズムについては、その内容を厳密に追及し、不公正がないことの立証責任を提供者側に課した上で、かつ、不公正を埋め込んだ場合のかなりの額のペナルティを課す契約を締結した上で、利用してもよい。しかし、その場合も、動作基盤は、IaaS でなければならない。PaaS、SaaS では、事業者の側で、AI サービスに対してさまざまな偏向を加えることが容易で、「改良」という建前でこれを合法的に行なってしまえるリスクがある。PaaS、SaaS では、ユーザーとの利益相反意図がある事業者による AI 偏向攻撃に、ユーザーである国・自治体は、なかなか気付くことができない。IaaS であれば、事業者の側でユーザーとの利益相反意図である者であっても、ユーザーが支配管理する仮想ディスクイメージへの侵入と書き換えが必要で、それは難易度がとても高く、途中で発覚するから、そのような AI 偏向攻撃に対して堅牢となる。ただし、データセットやソフトウェアをアップデートする際に、そのような AI 偏向攻撃が入り込む可能性があるので、その際には注意を要する。それでも、ユーザーは、古いバージョンと新しいバージョンの差異を両方比較することができる。PaaS、SaaS の AI サービスのように、事業者によって何の前触れもなく AI サービスのエンジンやデータセットがアップデートされて、もはや古いバージョンと新しいバージョンの差異を指摘する機会すら奪われるという状況にならない。このように、IaaS 上で自ら AI ソフトウェアをインストールして利用することにより、事業者の側が偏向的 AI 変化を発覚のリスクなく加える安易な攻撃的行為を、PaaS、SaaS の AI サービスと比較して、効果的に予防することができる。この程度のセキュリティ上の警戒を常に怠らずにいれば、国や自治体における事務的な意思決定の補助ツールとして、安全に AI を利用することが可能となる。もし、このような警戒を怠ったならば、意思決定の権力構造に対して、AI 事業者による介入が可能となり、公正な意思決定が歪められるリスクがある。これは、民主制の過程を損なう原因となる。公務員は主権者である国民によって間接的にでも選任された者であるが、AI 事業者は国民によって選任された者ではないためである。AI の利用にあたっては、これを防止するための予防が不可欠である。

第 2 章　「自由なシステム」の提案

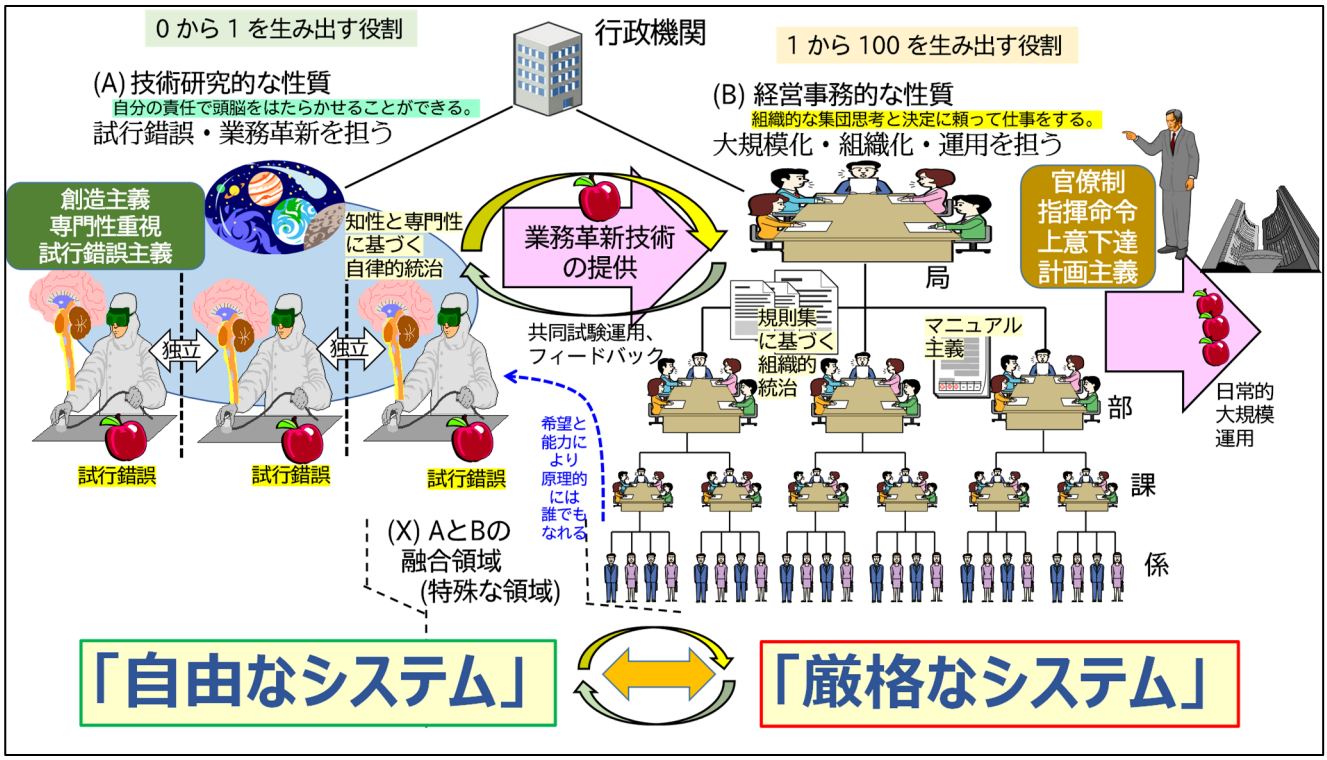
[redacted]に記載のあ[redacted]にある「人材育成」 (ネットワーク・セキュリティエンジニア)、「若手含む官民人材の参画」、「新技術 (開発) 」、「国際競争力強化」、「地方自治体 (の主体性の実現) 」については、2 つのレベルが存在する。95% の初級・中級的水準と、5% の上級的水準である。95% の部分については、[redacted]」 (厳格なシステム: 第 1 章) で作られるシステムの中で時間をかければ実現可能であろう。そこで、重要なのは、残りの 5% の水準の方々を開拓し、そういった方々の力を結集して、問題を解決し、そのための新技術を生み出し、高度な人材育成を実現することである。この 5% の領域を開拓するために、「自由なシステム」が必要である。

下記では、「自由なシステム」の概念を説明し、具体的なイメージ案を一応提示する。詳しくは、下記で述べる。

第 1 節　概説 ― デジタル庁における 2 つの役割の両立

1　概説

そもそも、デジタル庁は、国全体 (地方自治体を含む) の IT ネットワークおよびシステムインフラの構築にあたっては、2 つの役割を有している。



「自由なシステム」と「厳格なシステム」

(1)　「厳格なシステム」: 国の情シス ― 安全・確実なマイグレーション

これは、急ぎの仕事である。2030 年頃までに完成をさせることである。既存の省庁の NW、自治体の LGWAN を、安全確実に移行して延命させることである。セキュリティは表見的・短期的に実現されるが、それは、本質的に安全なものではない。しかし、緊急避難的には、この方法もやむを得ない。

(2)　「自由なシステム」: 国の進化 ― 新技術の研究の行政的土壌

これは、(1) よりもゆっくりとした仕事である。優秀な人材、多様な組織 (省庁・

独法・地方自治体)、日本全国に広がった広域な環境を活用して、たとえば、米国の歴史的な IT 技術 (コンピュータやインターネット、OS、クラウド技術等) を生み出得る程度の豊富な人数の IT 技術者・IT 研究者が、自律秩序的自由の元に活躍できるようにし、多数の問題を解決してゆき、官民併せて、各種の技術開発が多様に並列して試行錯誤するようにして、長期的なセキュリティの実現、人材育成、国際競争力を実現する。

上記 (2) の「自由なシステム」は、(1) の「厳格なシステム」と同等に重要である。(2) は、日本の役割を日本が果たすために重要である。欧米で製品化された不完全な技術は、日本に 30 年以上遅れてやってきて開花し、高品質に大成する。

半導体: 米国 1950s ⇒ 日本 1980s
コンピュータハードウェア: 米国 1950s ⇒ 日本 1980s
テレビゲーム: 米国 1960s ⇒ 日本 1990s
より古くから、造船、鉄鋼、化学、家電、自動車等も同様であった。

現代型 OS 群 (モダン OS、クラウドシステム、ネットワークシステム、セキュリティシステム) をみてみよう。これらは、2000 年代に米国で製品化された。

⇒ したがって、2030 年代に、モダン OS、クラウドシステム、ネットワークシステム、セキュリティシステムが、日本で開花し、高品質に大成する。

シリコンバレーの現代の技術者集団は、システムソフトウェア技術を強力な基礎として有していて、その上にアプリケーション層も立ち上げ、その二面で同時に活躍している。

⇒ 2030 年代に、日本版シリコンバレーが誕生し得る。システムソフトウェア技術を強力な基礎とし、その上にアプリケーション層も立ち上げる。日本人技術者集

団は、その二面ともに世界中で中心的責任を果たし、米国発祥のソフトウェア技術を安定的に完成させる。

日本は、全く新しい技術を生み出すというよりも、どちらかというと、既存の外来技術を磨き上げ、高品質化・高信頼性を実現することが得意である。日本人が生み出すものは、それほど、画期的という訳ではない。だいたいは地味な話である。しかし、世界中で最も高品質で優れたなものとして、すすんで選択されるのである。

以下は、全世界的な爆発的需要の確実性が高いものの一例である。これらを日本人は作ることができるようになる。これらを作るための技術的基礎体力を、日本人集団は、実はすでに身に付けているのである。(2) の「自由なシステム」からは、行政上の問題を解決しようとする過程において、以下のような技術が生まれるであろう。

ア. 新たな安全なパブリッククラウド技術

- 一極集中問題の解決
- マルチクラウドを安全簡単に分散利用できる仕組みの実現【第 3 章参照】
- 現在の外資系パブリッククラウドサービスと同等のサービスを誰でも構築できる技術
- 2030 年～2040 年代までに発生し得る深刻なクラウド・ショックを予防・解決

イ. 重要システムにおけるシステムソフトウェアを数十年間安全に運用できる技術

- EoL となった古い Windows / Linux、データベース等を 40 年間保たせる技術
- 開発元がメンテナンスせず、ソースコードも非公開の購入済みソフトウェアを、ユーザーが自由・簡単に改造してゆく技術

ウ. 100 年間安定して使えるコンピュータネットワーク技術や装置

- PC における Windows のような存在、スマホにおける Android のような存在の、ネットワーク機器バージョン
  今は、ネットワーク機器とソフトウェアは強度に結合してしまっている。どのベンダでもネットワーク機器に組み込んで出荷でき、アプリ・マーケットも有するネットワーク OS が日本から登場するであろう。

上記 (1), (2) の役割は、時間軸・目的・価値・人材の水準が異なる。長期的には単一の目的であっても、当初は、異なるものである。(1), (2) を同一視して議論しようとする限り、前進は困難である。(1) は厳格で計画的な思想、(2) は自由な思想を基礎として、両方を同時並行に進めることになる。

「デジタル敗戦」という用語がしばしば政府関係者 (特に政治家) ・報道機関によって用いられているが、上記 (1), (2) の概念が異なることを理解されていないことが原因の混乱がみられる。米国・中国などの IT 技術立国に対比した遅れを意味する (2) を解決すること、日本版シリコンバレーの出現を願うことこそが、「デジタル敗戦」という言葉に集団的に込められている高水準の意義である。ところが、そのためにいくら (1) の方向性のみ考えても、これは、低水準の意味 (例えば、「行政電子申請 Web アプリが貧弱である」等の意味) だから、話が全然噛み合わないのである。これを理解した上で、的確な方向性を提示できた政治家は、日本中の技術系・人文系の両方の人材から評価され、トップクラスの支持を得られる。

さて、われわれは、今や (1), (2) の 2 つの性質の違いを正しく認識するに至った。これから、(1), (2) の一方をもう一方に包含することなく、分散並行して両刀的に進めれば、必ず、日本は、すでに一位に輝いた各種の産業と同様に、IT 技術においても、再び世界一位の地位を譲り受けることができるのである。

2　デジタル技術を生み出す上での日本の役割

日本の世界におけるこれからの重要な役割は、脆い不確実な現代世界の IT・デジタル環境を、日本製技術によって改良し、世界における IT の長期的平和・安寧を実現することである。

これまでの IT・デジタル環境を支えるシステムソフトウェア技術は、実験的・

短絡的に技術者の好奇心本位で生み出されている、未成年の不良学生のようなものである (攻撃的で短気的な米国 IT 事業者とその技術者集団の行動を見よ)。

これが、成人として安定成長し、やがて社会を真に安定して支える程度に至るには、日本が、米国由来の IT 技術を完成させ、平和・中立的な供給者として全世界に普及させ、世界の IT 化に貢献する必要がある。

日本人の IT 技術者集団と事業者集団は、未だその存在が世界的に認識されていないけれども、水面下で秘かに蓄積増大している、21 世紀最大級の、驚異的な氷山下に韜晦(とうかい)する楽隊のようなものであり、世界中はその出現に震撼するであろう。

3 日本の IT 技術発展における行政機関の役割

既製品に頼らず、官僚と技術者たちが一生懸命問題解決・技術開発に取り組むことで、良い技術が埋まれ、これが世界を塗り替える。これは、歴史的に明らかである。

(1) 米国の歴史

米国の歴史をみると、たとえば、行政機関の課題である大規模計算処理・大規模通信処理を実現しようとして、既製品に頼らず、官僚と技術者たちが一生懸命技術開発に取り組んだ結果、現代型コンピュータシステム (ロスアラモス研究所でのノイマン型コンピュータの開発)、現代型インターネット (国防総省での研究用コンピュータ 3 台の接続が発端) などが出現している。

その上で、巨大企業の課題である事務処理 (文書処理) を実現しようとして、既製品に頼らず、社員たちが一生懸命技術研究に取り組んだ結果、現代型 OS である UNIX (AT&T 電話会社における文書処理システム開発が発端) が出現している。

さらに、その上で、巨大企業の課題である大量コンピュータリソース管理を実現しようとして、既製品に頼らず、社員たちが一生懸命技術研究に取り組んだ結果、現代型クラウドである AWS (Amazon Web Services) が出現している。

(2) 日本のこれからの現象

これと同じことが、日本でも発生する。

日本の行政機関は、巨大な情報処理の問題を抱え、大規模なコンピュータネットワークと優秀な人材を擁し、リソースに溢れ、これらを解決しなければならない状況にある。

よって、歴史の法則に基づき、既製品に頼らず、官僚と技術者たちが一生懸命問題解決・技術開発に取り組むことで、良い技術が埋まれ、これが世界を塗り替えるであろう。

(3) その後の歴史

日本人は、現在全世界で不足している、高品質なシステムソフトウェアの構築手法 (たとえば、どのようにすれば、Windows や AWS のような高度複雑なシステムソフトウェアの主要部分が作れるのか) を体系化し、日本語文献にまとめていく。これらは現在米国企業の現存技術者集団の頭脳に存在する秘伝のタレのようになっていて、彼らとしてもいまいち体系化していないから、大いに価値がある。日本人の作り上げた体系化文献群は、英語化・アジア語化され、アジアの国々の方々もやがて同じようなものを作り出すことができるようになる。

第 2 節　「自由なシステム」の方針案の提案

(1)　「自由なシステム」とは

自由なシステムは、優秀な人材、多様な組織 (省庁・独法・地方自治体)、日本全国に広がった広域な環境を活用して、たとえば、米国の歴史的な IT 技術 (コンピュータやインターネット、OS、クラウド技術等) を生み出得る程度の豊富な人数の IT 技術者・IT 研究者が、自律秩序的自由の元に活躍できるようにし、多数の問題を解決してゆき、官民併せて、各種の技術開発が多様に並列して試行錯誤するようにして、長期的なセキュリティの実現、人材育成、国際競争力を実現するためのシステムである。

それでは、自由なシステムは、どのような方式で設計すればよいだろうか。

(2)　自由なシステムを構築・運営・使用する行政職員の特性

5% の水準の人材は、ゼロトラストシステム風製品を研究することはあっても、日常的にそれらに依存しない。自ら、より良いゼロトラストシステムを開発する。クラウドシステム製品を研究することはあっても、日常的にそれらに依存しない。自ら、より良いクラウドシステムを開発する。ネットワークシステム製品を研究することはあっても、日常的にそれらに依存しない。自ら、より良いネットワークシステムを開発する。今の、または将来の LGWAN や GSS ネットワークを日常業務で頼ったり、課題を研究することはあっても、日常的にそれらに依存しない。自ら、より良い、さらに次世代の GSS ネットワークや LGWAN を開発する。そういった人々が、2030 年以降の日本の行政システムを作り出し、技術を開発し、それらの技術は社会還元され、日本から製品となって全世界に出て行く。外国製の電子技術や家電や自動車から 30 年遅れて、日本から良い半導体、家電、自動車が作り出されたのと同様に、2000 年代に外国で始まったクラウド革命から 30 年遅れて、日本から 2030 年代に良い IT システム技術が出ていくことになる。これは歴史的な法則である。そして、米国のコンピュータとインターネットの歴史のとおり、それらの創出にあたって重要な主体的役割を担うのは、国の行政部門と大企業の事務部門における問題解決である。行政部門・事務部門の問題を解決しようと

した若手技術者・研究者・官僚たちが作った技術が、コンピュータ、インターネット、UNIX、クラウド技術として生まれている。日本においても、これからそれが発生する。その主体となる若手人材は、前述の 5% の職員の中に埋もれているのである。今 20 代あるいは 10 代の方々 (が就職した後) である。これらの特殊的な人材に対しては、自由なシステムが必要である。自由なシステムは、年長者が計画した上で、それらの行政職員に対して与えられるものではない。それらの行政職員たちの手によって作られるものである。年長者の役割は、一番始めの部分を立ち上げて、若手人材にバランスのとれたリスク管理と秩序の整った自由を提供し、後見人的に保護をすることにある。

(3) 自律実験合同ネットワークの必要性

そういった自由なシステムを実現するためには、どのようにすれば良いか。自由なシステムに関わって試行錯誤を行なう 5% の人材が、比較的安全かつ秩序だった自由の元に取り扱うことができる、基本的構成要素を色々と用意してみながら、そのようなものに興味を見出すような各行政主体 (中央省庁、地方自治体、独立行政法人、J-LIS 等) の職員のうち特に水準の高い若手人材をうまく見つけてきて、組織を超えた自治的管理体を、合同的に作り出すのが良いと考えられる。ここで仮にこれを自律実験合同ネットワークと呼ぼう。この合同的な仮想の組織体は、人的および物的な意味で、ネットワークである。合同ネットワークは、組織の壁を越えて合同で利用可能な実験目的のコンピュータ・ネットワークを構成するであろう。そのコンピュータ・ネットワークは、5% の高度人材に特化された、かなり自由で、強力で、研究開発や自由実験が可能で、よりセキュアなネットワークである。また、合同ネットワークは、組織の壁を越えて合同で相互協力可能なさまざまなプロジェクトをホストする人的ネットワークを構成するであろう。その人的ネットワークは、自分の頭脳で高度複雑な事柄をいつでも学習し、自由に問題解決する能力を有する 5% の物好きな若手行政的人材によって構成され、IT に関するさまざまな問題を自らの問題として解決しようとし、長期的視点で技術開発と人材育成が行なわれる。この合同ネットワークが、いずれの組織にも強度に依存せず、全組織が主体であるような形で、各組織の空白的中央領域部分に自然に結成がなされるように、

うまく誘導する必要がある。合同ネットワークは、行政的な所与のものではなく、若手人材によって、自然に、各組織の間の中間空白領域に、自然に結成がなされたように見えなければならない。特定の組織が中央部分を担うと、それ以外の組織の人材からみると主体性に対して距離が感じられてしまう。IT において日本の行政系組織の縦割り状況下でプロジェクトがいまいちうまくいかないのは、いずれかの組織を主として、他を従とするような構造で作ってしまうため、主となる組織の人々にとっては面白いと感じられても、従となる周辺組織の人々にとっては一種、他人事として感じられる点にあると考えられる。それは防ぐ必要がある。

(4) 合同ネットワークは、あまり目立たずにいたほうがよい

そして、合同ネットワークは、行政的には、一応の暗黙的承認がなされているだけで十分であり、行政的に何か計画し、予算措置も講じ、発注して用意するという必要はない (それでは「厳格なシステム」となってしまう)。合同ネットワークには予算を多量に付ける必要もない。当面の間は、明示的な予算措置すら不要である。予算措置は、合同ネットワークが相当に良い成果を挙げてからでよい。しばらくの間は、合同ネットワークの各所は、各行政主体組織が参加人員をして持ち寄りで余剰資材と時間とで少しずつ構築されていくことになる。各部分は各参加人の主体的責任と役割で構築される。合同ネットワークは、うまくいくかいかないかわからない。これがうまくいく秘訣は、一応の行政的承認 (黙認) がなされていて、細かいことは年 1、2 回くらいどこかに (このような会議体に) 報告される程度でよく、トップマネジメントからみた過度な期待が加わらないようにすることである。合同ネットワークは、素晴らしいものであるので、もし目立って過度な期待が生じたならば、すぐさま、自民党の会議などで紹介されて、これは大変素晴らしいから、どんどんとやりたまえ、という風な模様になってしまうであろう。しかし、それでは、急いで物事を決めないといけないという圧力が生じて、行政的意思決定の解決がなされ、本来選択されるべき長期的選択肢よりも短期的な目先の目的が優先されてしまう。合同ネットワークは、別段何か納期や約束目標があるわけではないし、各参加主体は本業の合間に少し息抜きでこれに携わるという程度で良いのであるが、目立って期待をされると、いつの間にか、合同ネットワークが本業のようなタイプの

せかせか管理者が出てきて、その管理者のペースでプロジェクトマネジメントなど開始され、ガント・チャートなどが出現するようになるのである。そうなってしまったら、合同ネットワークは、もともとの「自由なシステム」を離れて、いよいよ、「厳格なシステム」の性質を帯びてゆくような危険な状態に陥る。そうすると、若者人材たちは面白くないと感じて、合同ネットワーク（「合同ネットワーク１」と呼ぼう。）から離れてゆき、新しい「合同ネットワーク２」など作ろうとすることは、間違いがないことである。結局はたいていのプロジェクトや事業体の崩壊はこのようにして繰り返される。それもまた新陳代謝である。シリコンバレーの各企業の歴史をみれば明らかである。しかし、せっかくやるのであれば、長持ちをさせたほうが面白い。だから、合同ネットワーク１、２、３、・・・ と数年ごとに消失しては新規作成されるよりも、何十年か合同ネットワークが持続するようにしたいと考える。そのためには、十分な成果が出るまで、目立たず、期待もさほどされない中で、予算も使わず、行政的黙認的体制の中で、ゆるやかに進化していくうまい生態系を作ることが重要である。

(5) 自由なシステム ― 合同ネットワークの具体的な内容

さて、上記のように自由なシステムの発想の実装イメージについて記述したが、これは相当抽象的なものである。なぜならば何か行政的計画をすると、それは当事者たち若手にとっては所与のものとなり、自由なシステムの発想から遠ざかるリスクがあるためである。しかし、それでは何のことやら全く不明である。こういうことを提起する際には、一応責任を持って、具体的に何か記述したほうが分かりやすい。というわけで、今流行のゼロトラストやネットワーク、クラウド等に関連して、[redacted]に倣って、以下のように仰々しく書いてみると、下記のようになった。

自由なシステム ― 合同ネットワーク
(ゼロトラスト)

2030 年以降を目標とした長期的視点で、既存のゼロトラストソリューションよりも安全で安定し、行政機関のような高度複雑環境において適応するシステムを生

み出すことを目指して、ゼロトラスト的思想を有する各種システム、ゼロトラスト的システムを実現するためのフレームワーク技術といったセキュリティ技術を、高度な能力水準を有する行政 IT 人材が自ら構築し、実験し、実証することができる環境を整備する (成果物は、2040 年頃の次々期国・自治体のアプリケーションおよび端末管理の基礎となるであろう)。

(ネットワーク)

2030 年以降を目標とした長期的視点で、既存のネットワークソリューションや現代の GSS や LGWAN よりも高速、低遅延、安全、自由、柔軟なネットワークを生み出すことできる環境を整備する (成果物は、2040 年頃の次々期 GSS や LGWAN の基礎となるであろう)。

(クラウド)

2030 年以降を目標とした長期的視点で、既存のパブリッククラウドを行政的に利用する際の各種問題 (コスト、性能、完全性、機密性、可用性) を解決する新たなクラウドシステムを生み出すことできる環境を整備する (成果物は、2040 年頃の次々期ガバメントクラウドの基礎となるであろう)。

基本的考え方

複数の行政主体 (中央省庁、地方自治体、独立行政法人、J-LIS 等) の職員のうち特に水準の高い若手人材をうまく見つけてきて、それらの人員がユーザーとなり同時に管理者となって、人的ネットワークと物的コンピュータネットワークを構築してゆく。この国地方ネットワーク会議のような行政的会議からは黙認的承認と後見を行なうにとどめ、予算は当面つけず、2030 年以降に成果が出ればよいという長期的な視点によって自然的発育を目指す。

人材育成は、技術者に留まらない。成果物は、技術に留まらない。良いシステム統治体制を有する人材や方法の育成も、重要な成果物である。参加者は、ユーザーの観点も、システム管理者の観点も、技術者の観点も、組織経営者の観点も、いずれも身に付けて、これらを調和させることが期待される。自由なシステムの統治を行なう側と、統治を受ける側は、同一である。自らの集団を統治する安全で自由な

ルールを発見し改良するのである。

(6) 「厳格なシステム」と「自由なシステム」との関係

自由なシステムは、厳格なシステムと、時間軸・目的・ユーザー水準が全く異なる。したがって、互いに競合・競争することはない。自由なシステム側が厳格なシステム側の方針に干渉することはなく、その逆もまた存在しない。いわば一定の暗黙の和平条約のようなものが形成されて、互いに相手方に対して口を出さないということで、両方とも静謐を享受できるのである。

それでは、自由なシステムと厳格のシステムとは、完全に独立され互いに隔離された要素なのであろうか。決して、そうではないのである。第一に、自由なシステムは長期的視点で役立つ成果物を生み出し、これは、厳格なシステムに取り込まれていく。5% の作った成果物は 95% でスケールしてこそ価値がある。厳格なシステムはこれにより利益を受ける。また、行政主体の 95% でスケールした技術は、全世界で役に立つことが実証される。日本全体がこれにより利益を受ける。第二に、自由なシステムの構成員は、厳格なシステムにおいて発生している技術的・運用的問題を参考にして、はじめて、自由なシステムの側を発案・改良できることもあるのである。これは、技術上の国際競争力を生み出すために極めて重要な利点である。技術開発者と、ユーザー組織とが、信頼関係のある一体的関係にあれば、技術開発者はユーザー組織の内部で発生している既存技術の問題を的確に把握し、いちはやく改良方法を発案することができる。通常はこの 2 者は分離されているので、ユーザー組織からのフィードバックに頼るしかない。組織の壁によって、このフィードバックは情報量が欠落してしまい、遅延が発生する。しかし、自由なシステムを構成する職員たちは、公式な人的・組織的には、厳格なシステムを構成する職員たちの中に包含されている。前記の情報量の欠落と遅延の発生がほとんどない。これは、素晴らしいことである。日本がこれから国際競争力のある良い IT 技術を生み出す不可欠の鍵となるのである。

さて、この自由なシステム ― 合同ネットワークをどのように進めればよいか。

この会議に参加されている方々を含めて、水準の、色々な方々のアイデアをいただきながら、また、参加してやってもよいという物好きの方々には参加していただいて、結構適当に始めるべきである思われる。前記のとおり、これを明確に計画的に始めてしまうと、いつの間にか、自由なシステムは、厳格なシステムに変質してしまうリスクがある。したがって、自律的・自然的に立ち上げる必要がある。到底、この国・地方ネットワーク会議の 4 回の短い期間に議論することは困難な事柄である。だが、自由なシステムの実現は、この「国・地方ネットワークの将来像及び実現シナリオに関する検討会」の掲げる重要な目標を実現するためには不可欠であると考えられる。

厳格なシステム (2025 年～2030 年) と比べて、自由なシステムは、別に何ら期限がある訳でもないし、立ち上げに失敗したところで、何かシステムが止まってしまうわけではない。もともとゼロだったものがゼロのままだから、損失はない。だから急いでやる必要はない。しかし、この急いでやる必要はないという考え方は、また曲者でもある。おそらく先人達もそのように考えて、いつの時代も、期限が差し迫った厳格なシステムのほうを大いに優先させてきたのであろう。だから、現時点で、行政主体を中心とした上記のような自由なシステム思想に基づく合同ネットワークは、誰でも考えつくものではあるのに、1 つも存在しない (結成されたこともない) のである。これではいつまでも自由なシステムが始まらないのである。という訳で、結局は、日本の歴史においては、いつかの時代かの、いずれかの集団が、少しくらいは、努力をして、自由なシステムを一念発起して起動しなければならない。まあ我々がやらなくても、後の 10 年後の世代が立ち上げるかも知れないが、立ち上げないかも知れない。いつまでも立ち上がらないかも知れない。だがせっかく我々はそういうことができそうな機会を得ているのだから、ここは後世に任せるのではなく、今立ち上げたほうがよい。うまくいけば、日本の IT の発展史における重要な歴史の流れを作り出したのはこの瞬間であるということになるであろう。という訳で、どのように開始するべきか、是非アドバイスをお願いしたい。